Navy Personnel Research and Development Center
San Diego, California 92152-6800 TN-91-12 April 1991

Troubleshooting Assessment and Enhancement (TAE) Program: Design, Development, and Administration

Harry B. Conner
Cathryn Poirier
Randy Ulrich
Thomas Bridges

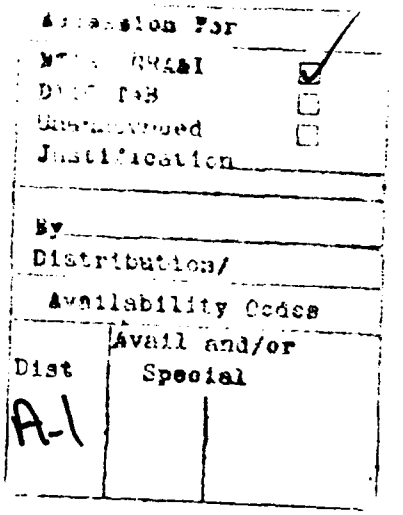
NPRDC-TN-91-12
Volume 1

April 1991

Troubleshooting Assessment and Enhancement (TAE) Program: Design, Development, and Administration

Harry B. Conner
Navy Personnel Research and Development Center

Cathryn Poirier
Randy Ulrich
Thomas Bridges
System Engineering Associates
San Diego, CA 92109



Approved and released by
J. C. McLachlan
Director, Training Systems Department

Navy Personnel Research and Development Center
San Diego, California 92152-6800

Identification of specific equipment and software is for documentation only and does not imply endorsement.

c-tree is a trademark of FairCom.

Corvus, Corvus Systems, PC/NOS, Netview, and Transporter are trademarks of Media Cybernetics, Inc.

Dr. Halo is a trademark of Media Cybernetics, Inc.

IBM and IBM PC are trademarks of International Business Machines Corp.

Meta WINDOW and MetaWINDOW/PLUS are trademarks of Metraphics Software Corporation.

Microstat, Microsoft, MS-DOS are trademarks of Microsoft Corporation.

PANEL and PANEL Plus are trademarks of Roundhill Computer Systems Limited.

ZDS Microstat is a trademark of Zenith Data Systems.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATE COVERED
	April 1991	Technical Note--Oct 87-Mar 90

4. TITLE AND SUBTITLE
Troubleshooting Assessment and Enhancement (TAE) Program: Design, Development, and Administration

5. FUNDING NUMBERS
0603720N-R1772-ET01

6. AUTHOR(S)
Harry B. Conner. Cathryn Poirier, Randy Ulrich, Thomas Bridges

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Navy Personnel Research and Development Center
San Diego. California 92152-6800

8. PERFORMING ORGANIZATION REPORT NUMBER
NPRDC-TN-91-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Chief of Naval Operations (OP-11)
Navy Department
Washington, DC 20350-2000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Volume 2 contains the TAE delivery software on diskettes.
For additional information, see NPRDC TN-91-11 and NPRDC TN-91-13.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of the Troubleshooting Assessment and Enhancement (TAE) R&D effort was to develop, test, and evaluate a low-cost, microcomputer-based system to provide an objective measure of the troubleshooting proficiency of Navy technicians. This technical note presents the results of the design and development effort. It provides the system requirements, program design description, TAE software, and the TAE administration guide.

14. SUBJECT TERMS
Troubleshooting, simulation, performance evaluation, troubleshooting training

15. NUMBER OF PAGES
153

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

FOREWORD

The troubleshooting Assessment and Enhancement (TAE) Program (previously titled Troubleshooting Proficiency Evaluation Program, TPEP) was sponsored by the Deputy Chief of Naval Operations (OP-11) and was performed under 0603720N-R1772-ET01. The purpose of the TAE program was to develop a low-cost, microcomputer-based system to provide an objective measure of the troubleshooting proficiency of Navy technicians.

This is the second of three technical notes that document the TAE program. The first technical note presents the results of the literature search, a methodology for developing a troubleshooting proficiency evaluation system, and the test and evaluation plan for the initial TAE system (Conner & Hassebrock, 1991). This technical note consists of two volumes. Volume 1 addresses the design and development of the computerized delivery system and a TAE administration guide. Volume 2 contains the TAE delivery software on diskettes.

The final technical note presents the results of the test and evaluation as well as the conclusions and recommendations for enhancing the TAE delivery system (Conner, Hartley, & Mark, 1991).

J. C. McLACHLAN
Director, Training Systems Department

SUMMARY

Problem

The Navy has limited means of measuring the troubleshooting proficiency of Navy technicians and their ability to contribute to operational readiness. Also, there is limited capability to maintain or enhance troubleshooting skills aboard ships or at Reserve Readiness Centers or to evaluate overall troubleshooting capability. As a result, there is limited ongoing feedback to the training command to improve the schools responsible for troubleshooting skills training.

Purpose

The purpose of the Troubleshooting Assessment and Evaluation (TAE) program was to develop a low-cost, microcomputer-based system to provide an objective measure of the troubleshooting proficiency of Navy technicians. This report documents the design and development of the TAE delivery system.

Approach

The approach was to construct a valid TAE test that would adequately sample the troubleshooting domain. Using an "expert" model, a set of troubleshooting evaluation factors was developed. These factors were incorporated into the design and development of the computerized TAE delivery system. The test and evaluation plan was designed to assess the TAE system, validate the TAE troubleshooting episodes, and assess the reliability and effectiveness of the episodes in evaluating performance of troubleshooting technicians.

Results

The TAE effort produced a troubleshooting proficiency demonstration system for the maintainers (NEC ET-1453s) of the Naval Modular Automated Communications System (V)/ Satellite Communications (NAVMAC (V)/SATCOM) hardware. The design and development of the TAE delivery system are described in terms of the system's functional requirements, the program design and installation, and administration requirements.

Conclusions and Future Efforts

The demonstration TAE system is currently implemented at the sites where the hardware training is conducted. Although there are no plans to modify or expand the current TAE capability, several recommendations for improvement are provided.

CONTENTS

INTRODUCTION

Problem

Currently the Navy has limited means to objectively measure the troubleshooting proficiency of shipboard technicians and their ability to contribute to operational readiness. Other than subjective supervisory opinion, there is no consistent and reliable way to assess the transfer of training, particularly hands-on training on hardware systems provided in Navy "C" schools. Once the "C" school graduate has been integrated into the ship's force, fleet commanders have no objective method to assess the technician's performance capabilities or skill degradation over time. In addition, the schools receive no quantifiable feedback identifying specific areas where troubleshooting training requires greater emphasis or improvement.

Due to limited availability of system hardware at "C" schools, actual hands-on training time is severely restricted. This limits the amount of time students explicitly use their system knowledge and, therefore, decreases the effectiveness of instructional programs. Once on-board, the ship safety hazards associated with corrective maintenance of weapon system hardware preclude the use of drill and practice exercises. This limits the technician's ability to maintain or improve troubleshooting skills.

Purpose

The purpose of the Troubleshooting Assessment and Evaluation (TAE) program was to develop a low-cost, microcomputer-based system to provide an objective measure of the troubleshooting proficiency of Navy technicians.

Specifically, the TAE program was to (1) assess personnel troubleshooting capabilities within the Navy training environment (e.g., "C" school and/or reserve training activities), (2) develop drill and practice for personnel in training awaiting hardware availability or active duty assignments, (3) improve curricula and training methods based on school troubleshooting assessment results, (4) provide fleet and reserve on-board training (OBT) through drill and practice exercises, (5) develop an objective measure of operational readiness of fleet ad reserve personnel in their area of systems hardware troubleshooting capability, (7) improve operational readiness, and (8) improve curricula and instructional methods as a result of operational fleet and reserve feedback of assessment/ evaluation data to the training community.

The TAE project resulted in a troubleshooting proficiency assessment demonstration for the high-technology (electronic/digital) maintainer community (NEC ET-1453) for the Naval Modular Automated Communications System (V)/Satellite Communications (NAVMACS (V)/SATCOM) hardware. This technical note documents the design and development of the TAE computerized delivery system. The test and evaluation will (1) assess the TAE troubleshooting evaluation and diagnostic factors, (2) validate the ability of the TAE episodes to evaluate and diagnose troubleshooting proficiency, and (3) assess the reliability and effectiveness of the TAE episodes to evaluate troubleshooting proficiency, diagnose results, and, thereby, lead to improved training.

Background

The TAE project was organized into three phases: analysis, design and development, and test and evaluation. The steps included:

1. Selection of the NAVMACS (V)/SATCOM hardware system and the NEC ET-1453 maintainer community for the demonstration.

2. Review of the literature to provide input into the design and development of the troubleshooting episodes and the test and evaluation procedures.

3. Design and development of computer software to support the evaluation program.

4. Design and development of the troubleshooting episodes selected as representative for the demonstration maintenance community.

5. Design and development of training assessment and training drill and practice episodes.

6. Design and development of a troubleshooting episode development capability to be used for other hardware systems.

7. Development of factors for evaluating troubleshooting proficiency.

8. Development of a test and evaluation plan stating the research hypotheses and analysis techniques.

9. Data collection, analysis, and reporting for the test and evaluation.

The TAE system computer software design and development efforts (Steps 3 through 6) are described in this technical note. The review of literature and a discussion of the theoretical and methodological issues in TAE design (Steps 1 and 2) are documented in Conner and Hassebrock (1991). The results of the test and evaluation (Steps 7, 8, and 9) are presented in Conner, Hartley, and Mark (1991).

APPROACH

Research Objectives

To develop the capability to discriminate between levels of troubleshooting proficiency, it was necessary to define the delivery system requirements within the TAE context. This effort proposed to use fleet subject matter experts and instructor ratings of TAE scoring profiles to construct a troubleshooting proficiency criterion. Once developed, this measure could be refined over time to produce a closer approximation of the ultimate criterion of troubleshooting proficiency. If the concepts of reliability and validity were established, then it would be possible to build a strong, logical connection between TAE and its ability to predict troubleshooting proficiency among electronics technicians in the fleet.

Whether TAE could be empirically validated as a predictive instrument for success in the "C" school program by using the various "C" school test scores as criterion measures was problematic at the time the delivery system was being designed and developed. However, it was assumed that the TAE test could accurately estimate actual performance capability at the time of "C" school graduation as well as predict subsequent fleet performance.

The manner in which the TAE test was constructed had to be logical and sensible. That is, the TAE test must have face validity. It was also imperative that the hardware system being used as the basis for the test was adequately sampled and developed into troubleshooting episodes (items). The troubleshooting episodes developed for the NAVMACS (V)/SATCOM hardware encompass the troubleshooting domain for the equipment and reflect actual equipment faults that technicians may encounter in the fleet.

Troubleshooting Evaluation Factors

The development of the evaluation factors utilized an "expert" model. That is, the experts in the field of endeavor under question generally define and stipulate the contents of the test (i.e., the test items or events), the relative value of the events, and the method of scoring. Given that the experts have determined the test to be given, the components of the test, and the method of weighing and scoring the events of the test, it is difficult to take issue with the ultimate results of individuals being measured by the standards established.

Development of the TAE troubleshooting evaluation factors followed a similar "expert" defined approach. Based on previous research findings (Conner, 1986, 1987) and inputs from subject matter experts, a questionnaire concerning factors related to the evaluation of troubleshooting skills was developed and disseminated to high-technology maintenance personnel in technical environments and in the fleet. Respondents were asked to complete a background information form and then rank order the factors in order of importance. Since the relative importance of the factors may change with conditions, the following conditions were assumed: (1) non-combat, (2) normal day in home port, and (3) fault was encountered during a normal systems check. The responses were tabulated and ranked.

A second questionnaire was developed and disseminated to subject matter experts to rank the factors according to their relative level of importance. The results were tabulated and converted to weighting factors to be used in evaluating an individual's performance. The final weights were the ones utilized in the TAE computerized scoring scheme for the test and evaluation. The TAE delivery system was designed such that the troubleshooting evaluation environment can easily be changed. Also, factors may be added, deleted, or modified and weights assigned to the various factors may be changed by the user.

TAE Episodes

Within the context of the TAE demonstration, troubleshooting is viewed as part of the corrective maintenance function. When a system is not functioning properly, corrective maintenance must be performed to return the system to an optimum operational state. Troubleshooting is the means by which the faulty components are identified. Once identified, the

faulty components can be repaired/replaced. Figure 1 displays this relationship. The TAE episodes were designed to measure the ability to troubleshoot by identifying the faulty component.

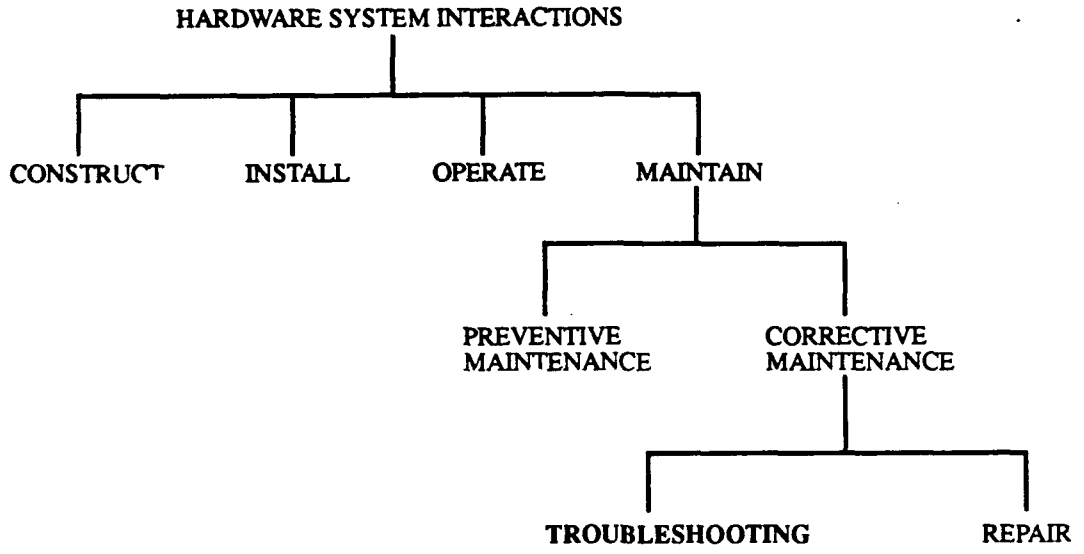


Figure 1. Hardware activity to troubleshooting.

The TAE testing format begins by displaying fault indicators. The subject uses a series of menus to review fault symptoms, front panels, maintenance panels, and diagnostic information; to select equipment; and to make reference designator tests or replace a Lowest Replaceable Unit (LRU). The subject's goal in the TAE test is to find the faulty LRU as defined by the maintenance philosophy of the system. This is done by selecting the suspected LRU for replacement. It is possible for the fault symptom to logically lead to an LRU that is not the faulty LRU as defined by the episode. This is indicated as a GOOD FAULT but not the specific faulty LRU.

The troubleshooting assessment episodes are listed below. No TAE episodes were developed for troubleshooting the TSEC/KG-36 due to the sensitivity and classification problems associated with this subsystem.

AN/USH-26 (V) Subsystem

1. Formatter A
2. Formatter B
3. Servo/Data
4. Parallel Interface
5. Control

AN/USQ-69 (V) Subsystem

1. Maintenance Panel Keyboard
2. Power Supply

3. CRT
4. 2nd, 3rd Page RAM
5. Micro Controller

AN/UYK-20 (V) Subsystem

1. Channel 16 Interface
2. Micro Channel 15 and IO Oneshot Control
3. Channel 14 Interface
4. Memory Interface
5. Memory Interface

CV-3333/U Subsystem

1. Sample Processor Assembly
2. Sample Data Generator Assembly
3. Spectrum Analyzer No. 2
4. Handset
5. Analyzer and Synthesizer Analog
6. Voicing and Channel Encoder
7. Pitch Analyzer
8. Spectrum Analyzer No. 2
9. Timing and Interface
10. Timing and Self Test

ON-143 (V)/USQ Subsystem

1. Level Converter
2. Transmit Sequence Control
3. Relay Card
4. Rec Synchronization
5. Red/Black Interface
6. Red/Black Interface Relay

RD-397U Subsystem

1. Punch Enable Signal
2. LD Signal
3. OD 3 Signal

TT-624 (V) 5/UG Subsystem

1. Input and Buffer Data Registers
2. Hammer Drivers
3. Paper Feed Control Logic
4. Output Decode
5. Serial Interface Logic

The TAE testing episodes developed for the demonstration may be used as troubleshooting training exercises as well as troubleshooting assessment tools. There are five demonstration systems at the Fleet Training Center, Norfolk and six demonstration systems at the Advanced

Electronic Schools Department, Service Schools Command, San Diego for training and evaluation purposes. Although the test and evaluation plan focuses on the ability of TAE to assess troubleshooting proficiency, TAE should also be viewed in the broader context as an instructional tool.

In addition to these testing episodes, three other levels of TAE episode presentation were planned: directive training, guided training, and test with feedback. However, only 14 directive training episodes were developed and no guided training episodes or tests with feedback were produced:

AN/USH-26 (V) Subsystem

1. Servo Data
2. Control

AN/USQ-69 (V) Subsystem

1. 2nd, 3rd Page RAM
2. Micro Controller

AN/UYK-20 (V) Subsystem

1. Card Location J06
2. Card Location A24

CV-3333/U Subsystem

1. Spectrum Analyzer
2. Synchro[illegible]zation, Control Logic

ON-143 (V) USQ Subsystem

1. Rec Synchronization
2. Transmit Sequence

RD-397U Subsystem

1. Punch Driver Assy
2. Reader Controller

TT-624 (V) 5/UG Subsystem

1. Output Decode
2. Serial Interface Logic

The directive training episodes are designed so that the student is, in effect, looking over the shoulder of an expert troubleshooter as a fault is discovered. The symptoms are provided and then information is presented on (1) what the symptoms should tell the troubleshooter, (2) what tests or checks should be made, and (3) what conclusions could be drawn from these tests or checks. Then, a test or check is accomplished. The results of the test or check are displayed, and the implication of that check or test are provided. This sequence is continued until the fault is identified. Throughout the sequence, the student observes the activity and follows the action in the technical

manuals (TMs). Information and graphics from the TMs are provided in the presentation as appropriate.

RESULTS

System Requirements

The TAE system was designed by a team of experts. Training specialists, subject matter experts, and software developers defined the requirements for development of a low-cost assessment tool which could also be used for training drill and practice. To lower costs, the system was to use off-the-shelf technology as much as possible. The intent of the project was to expand the TAF approach, not the utilization of hardware technology. The TAE troubleshooting episodes were designed to be low fidelity simulations to reduce the computer hardware and software programming requirements. However, within this construct, it was important to make the assessment approach as flexible as possible and to ensure that the delivery system was user friendly from the perspective of the students and the administrators. Appendix A presents the TAE System Requirements Document (SRD). The SRD provides a detailed description of the functional requirements of the TAE computerized delivery system.

Program Design

Appendix B provides a detailed description of the TAE software as it was designed and developed for the NAVMACS (V)/SATCOM demonstration. This information is intended to allow the user to expand or improve the current delivery system or modify the computerized program for a different hardware system. The program description is presented from the perspective of a software developer who is familiar with the concepts of software design. It includes data flow diagrams, data dictionaries, context diagrams, and other commonly used computer software tools, as documented in DeMarco (1978).

Delivery Software

Volume 2 (of this technical note) contains the TAE delivery software on diskettes. Appendix C provides instructions for installing the TAE software at activities that have the computers to run the program. Information is also included on the program structure and tools so that a qualified programmer can modify or improve the demonstration software, or use the programming as a point of departure in the development of a software package for a new hardware system.

Administration Guide

Appendix D provides the TAE Administration Guide that was developed for use in the test and evaluation of the NAVMACS (V)/SATCOM demonstration. There have been some improvements in the delivery system functions since the guide was prepared. These improvements are described in the TAE SRD (Appendix A) and included in the delivery software (Volume 2).

CONCLUSIONS AND FUTURE EFFORTS

At this time, the TAE system is implemented at the two sites where NAVMACS (V)/SATCOM hardware training is conducted: the Advanced Electronics Schools Department, Service Schools Command, Naval Training Center, San Diego, California and the Fleet Training Center, Norfolk, Virginia. The system has also been installed on the Mobile Pierside Trainers, Naval Station, San Diego, California.

Although there are no plans to modify or expand the current TAE capability, several recommendations for future improvements are provided below. Recommendations based on the results of the test and evaluation are reported in Conner, Hartley, and Mark (1991).

1. A networking capability would be useful in the classroom or Reserve Readiness Center environment.

2. The software should be made to recognize typographical errors in "Reference Designation" rather than to record them as out of bounds.

3. The episode authoring capability should be expanded so that subject matter experts can more readily develop and implement a wide range of troubleshooting exercises.

4. The analysis approach for the test and evaluation, which did not deal specifically with issues of interest to Navy "C" schools, should be expanded to evaluate issues related to curriculum standards, control, and improvement.

5. Additional TAE troubleshooting episodes to provide directive training, guided training, and tests with feedback should be developed. Then, a complete and comprehensive troubleshooting skill development, maintenance, assessment, and evaluation program would be available for personnel from the novice to expert skill levels for use by active duty personnel in the school or fleet environments and by reserve personnel at the readiness centers or aboard ship during active duty periods.

REFERENCES

Conner, H. B. (1986, October). Troubleshooting proficiency evaluation project (TPEP). In *Proceedings of Military Testing Association Conference*, Mystic, Connecticut.

Conner, H. B. (1987, April). Troubleshooting proficiency evaluation project (TPEP) for the NATO Seasparrow Surface Missile system (NSSMS). In *Proceedings of First International Manpower and Training Conference of the National Security Industrial Association*, Luxembourg.

Conner, H B., & Hassebrock, F. E. (1991). *Troubleshooting Assessment and Enhancement (TAE) program: Theoretical, methodological, and test and evaluation issues* (TN-91-11). San Diego: Navy Personnel Research and Development Center.

Conner, H. B., Hartley, S., & Mark, L. J. (1991). *Troubleshooting Assessment and Enhancement (TAE) program: Test and evaluation* (TN-91-13). San Diego: Navy Personnel Research and Development Center.

DeMarco, T. (1978). *Structured analysis and system specification.* New York: Yourdon.[1]

[1]Also cited in Appendix B.

APPENDIX A

TROUBLESHOOTING ASSESSMENT AND ENHANCEMENT (TAE) SYSTEM REQUIREMENTS DOCUMENT

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Overview

This appendix describes the functional requirements of the Troubleshooting Assessment and Enhancement (TAE) system, designed for the Naval Modular Automated Communications Subsystem (NAVMACS). It describes the system from the perspective of the development team's training specialists and is intended as the primary reference for the system's development. All system and design implementation decisions will be guided by the requirements noted herein. Identification of specific equipment and software is for documentation only and does not imply an endorsement.

TAE Objectives

The Navy must be able to measure objectively the troubleshooting proficiency of its technicians. There is no way to evaluate the success of on-board technical training or the effect of hands-on training in Navy schoolhouses. To address this problem, the Navy Personnel Research and Development Center (NPRDC) initiated a computer-based Troubleshooting Assessment and Enhancement (TAE) Program.

The primary objective of TAE is to develop a system that will evaluate the troubleshooting proficiency of Navy technicians. Figure A-1 presents a context diagram of the TAE system. The system's major components include software programs (TAE software), microcomputer workstations (TAE hardware), Navy troubleshooters and instructors, and lessonware authors (subject matter experts). Each TAE system component plays a role in achieving the system's objective. The TAE software performs three duties: It allows subject matter experts to generate simulated troubleshooting episodes, it presents these episodes to Navy troubleshooters and records their interactions, and it allows instructors to review and analyze these troubleshooting actions. The role of the TAE hardware is to host the TAE software in as wide a variety of Navy classrooms and ships as possible. The role of the subject matter experts is to generate the system's lessonware. The system's troubleshooters must exercise the TAE software that presents this lessonware, and the system's instructors must exercise the TAE software that assesses the troubleshooter's performance.

TAE Development History

The TAE project was started in 1982 and terminated in 1990. The acronym TPEP (for Troubleshooting Proficiency Evaluation Program) was used for the project until mid-1989, when it was changed to TAE to better reflect the project's objectives.

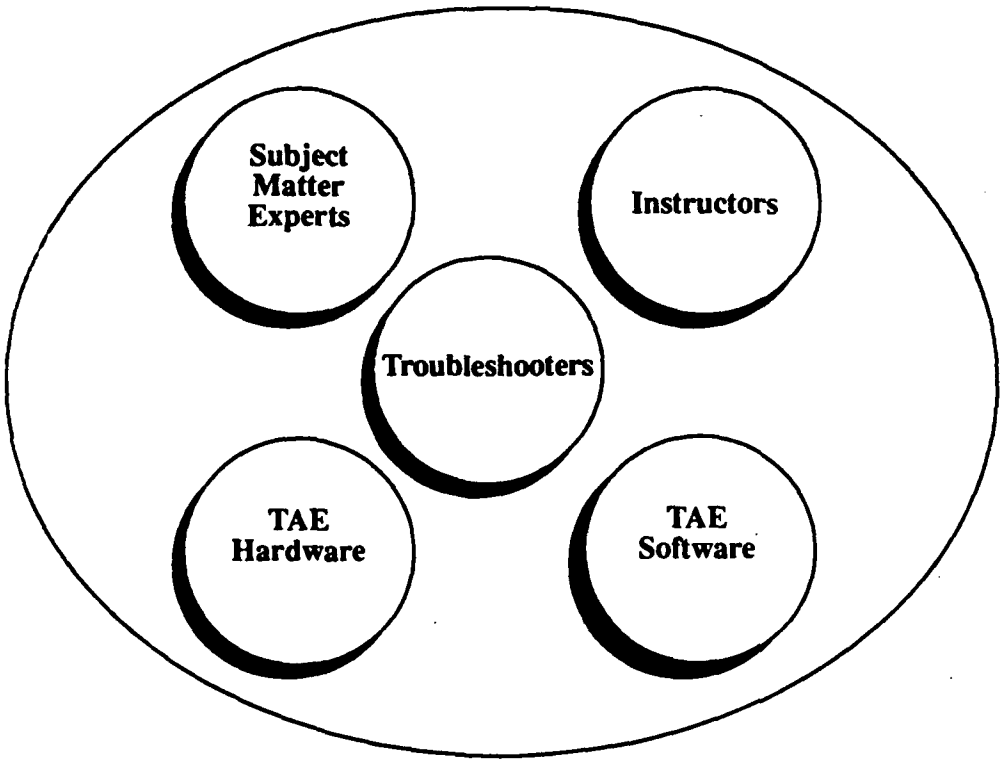


Figure A-1. TAE System Context Diagram.

A prototype version of the TAE system was developed on KAYPRO microcomputers in 1982. This prototype validated the system's concept by demonstrating a capability to assess troubleshooting performance using a low-cost microcomputer. The prototype was never distributed to Navy classrooms, since it was hosted on a non-standard microcomputer and contained several software flaws. In 1985, the Personal Electronic Aid for Maintenance (PEAM) project needed to determine if a new troubleshooting tool (the PEAM device) helped technicians troubleshoot the NATO Sea Sparrow Missile System. The success of the TAE prototype motivated the PEAM project to fund a partial implementation of TAE on IBM personal computers. This implementation corrected many of the software deficiencies of the KAYPRO prototype. Funding for the PEAM project ceased in early 1987 so many features of the software were not completed.

As the PEAM project phased out, a new TAE customer emerged: the Naval Modular Automated Communications Subsystem (NAVMACS). NAVMACS is an electronic system built from eight pieces of equipment. As part of their training in the Navy's Advanced Electronics School, Navy technicians must take a 12-week course in the operation and maintenance of NAVMACS. The NAVMACS project used TAE as an evaluation tool to measure the students' troubleshooting proficiency. At the end of 1987, a prototype version of the episode presentation software was completed. During the first half of 1988, several features and improvements were made to the NAVMACS TAE software. It was put into use in the classroom on a trial basis in mid-1988. Further enhancements were made throughout 1988 and 1989. Beginning in late 1988, the

NAVMACS TAE software was employed to collect student troubleshooting data for analysis. Analysis of the data was begun in mid-1989 and completed in early 1990.

NAVMACS TAE Description

NAVMACS is a communications system used to process message traffic between ships and the shore. It is built from eight pieces of equipment: a computer processor [AN/UYK-20(V)], a data terminal set [AN/USQ-69(V)], a printer [TT-624(V)5/UG], an analog-to-digital converter [CV-3333/U], two recorder/reproducers [AN/USH-26(V) and RD-397/U], a cryptographing device [TSEC/KG-36], and an interconnecting group [ON-143(V)/USQ]. When referring to these pieces of equipment, their model names [i.e., AN/UYK-20(V)] will be used. Figure A-2 provides a diagram of the NAVMACS system.

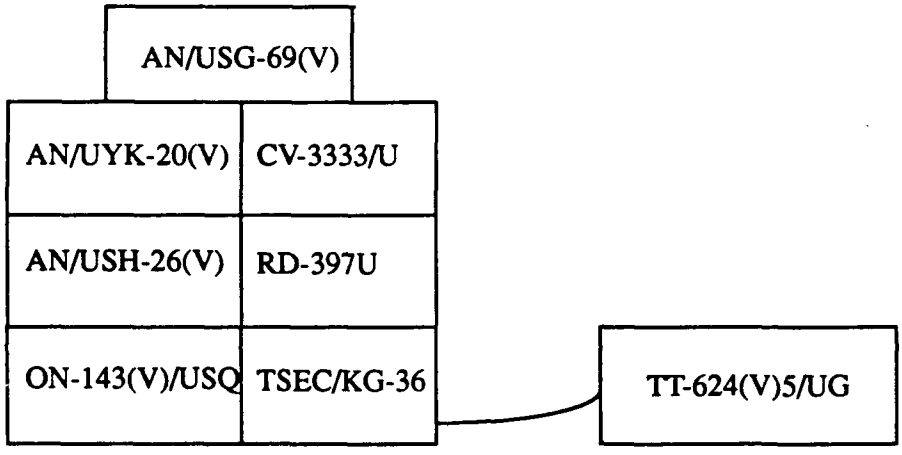


Figure A-2. NAVMACS System Diagram.

TAE simulates three levels of analysis when examining NAVMACS: the system level, the equipment level, and the component level. Figure A-3 illustrates these levels of analysis. From the system level, NAVMACS is one functioning entity, greater than the sum of its parts. All of the system's equipment interacts dynamically. Problems in one piece of equipment may produce abnormal symptoms in other equipment. From the equipment level, one of the eight pieces of equipment is selected for further analysis. Each piece of equipment can be viewed as a system within itself, unique from all other equipment. A piece of equipment is built from components. The component level is the lowest level of analysis TAE will simulate. Each component is identified by a unique name, called the component's reference designator. An example of a reference designator is "A1B1". Components may be replaced if they are determined to be faulty.

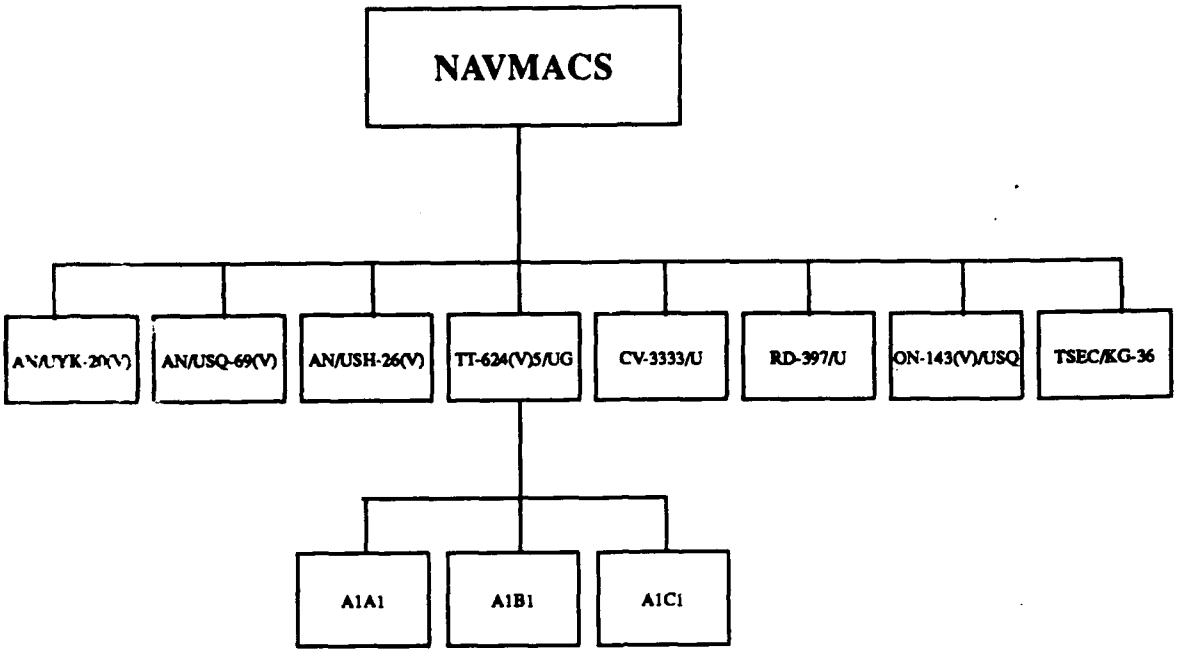


Figure A-3. Levels of Analysis for the NAVMACS TAE System.

When the NAVMACS system fails, the technician can perform a number of simulated tests to help troubleshoot the problem. There are different tests for the system level, equipment level, and component level.

Summary of TAE Functional Requirements

The functional requirements of the NAVMACS TAE system fall into the following six groups:

1. General Requirements - The ultimate high-level requirements the NAVMACS TAE system must meet.
2. Hardware Requirements - The host microcomputer required for the NAVMACS TAE, as selected by NPRDC.
3. Episode Authoring Requirements - Requirements the episode authoring software must meet.
4. Episode Presentation Requirements - Requirements the episode presentation software must meet.
5. Student History Viewing Requirements - Requirements the student history viewing software must meet.
6. Local Area Network Requirements - Requirements the NAVMACS TAE system must meet to operate under a local area network as selected by NPRDC.

The requirements in each group are described in detail in the subsequent sections.

GENERAL AND HARDWARE REQUIREMENTS

General Requirements

The NAVMACS TAE system must meet the following general requirements:

1. It must provide a low-fidelity simulation of NAVMACS.
2. It must be able to present 70 different problem "episodes" to NAVMACS technicians. Ten episodes will be developed for each piece of equipment, except for the TSEC/KG-36. (No episodes will be developed for the TSEC/KG-36 because its operation is classified.) The source of the problem for each of the 10 episodes for each piece of equipment will reside in that piece of equipment. The main elements of an episode include: the abnormal symptoms the NAVMACS system is displaying, a single faulty component that is the source of the problem, and the state of other equipment and components in NAVMACS relevant to the episode.
3. It must allow NPRDC personnel to add, modify, and delete the episodes.
4. It must collect Navy troubleshooters' interactions with the simulator and store the interactions in a student history database.
5. It must identify Navy troubleshooters by social security number.
6. It must track the episodes each troubleshooter has attempted.
7. It must evaluate troubleshooting actions as good, out of bounds, illogical, or invalid.
8. It must provide two categories of system level tests: Fallback and Load Operational Program.
9. It must provide two categories of equipment level tests: Front Panels and Within Bounds. Front Panel tests allow the troubleshooter to view a graphic image of any of the eight pieces of equipment. Within Bounds tests allow the troubleshooter to determine whether or not a piece of equipment is related to the problem.
10. It must allow troubleshooters to replace components in a piece of equipment, and then inform the troubleshooter whether the replacement fixes the NAVMACS system.
11. It must provide a variety of component-level tests, dependent upon each piece of equipment.
12. It must allow episode authors to identify certain tests as "proof points". A proof point is a test that provides a clue as to which component in the system is faulty.
13. It must allow episode authors to group proof points into "lone" proof points and proof point "pools".
14. The TAE software programs must be menu- and form-driven so as to ease the user's interaction with the software.
15. It must allow Navy instructors to display and print student histories.
16. It must allow Navy instructors to edit student histories.

17. It must allow Navy instructors to calculate, display, and print the scores for each student history.

18. It must allow Navy instructors to define the scoring values and weights for each of the scoring factors.

19. It must allow an instructor to convert the collected student history data into a format usable by the Microstat Version 4.0 statistical analysis software package.

20. It must allow an instructor to archive (save) the current student history database files under a given name, retrieve archived student history databases by name, and combine two different student history databases.

21. It must allow an instructor to view or print a student's final scores all together on one page. It should calculate the class average for each episode, and the student's average over all the episodes. It must also allow the instructor to view or print the class average over all of the episodes.

22. It must allow an instructor to delete a student from the student history database entirely.

Hardware Requirements

The host microcomputer for the NAVMACS TAE software is a Zenith 248 outfitted with:

1. A 360-kilobyte floppy disk drive
2. A 20-megabyte hard disk
3. 640 kilobytes of RAM
4. A keyboard
5. An Enhanced Graphics Adapter (EGA) card
6. A 13 inch EGA color monitor
7. A printer

EPISODE AUTHORING REQUIREMENTS

Overview

This section describes the functional requirements of the Episode Authoring System (EAS) for the NAVMACS TAE system. The EAS is also referred to as the Computer-Assisted TAE Episode Authoring System (CATEAS). Figure A-4 illustrates how the TAE EAS works.

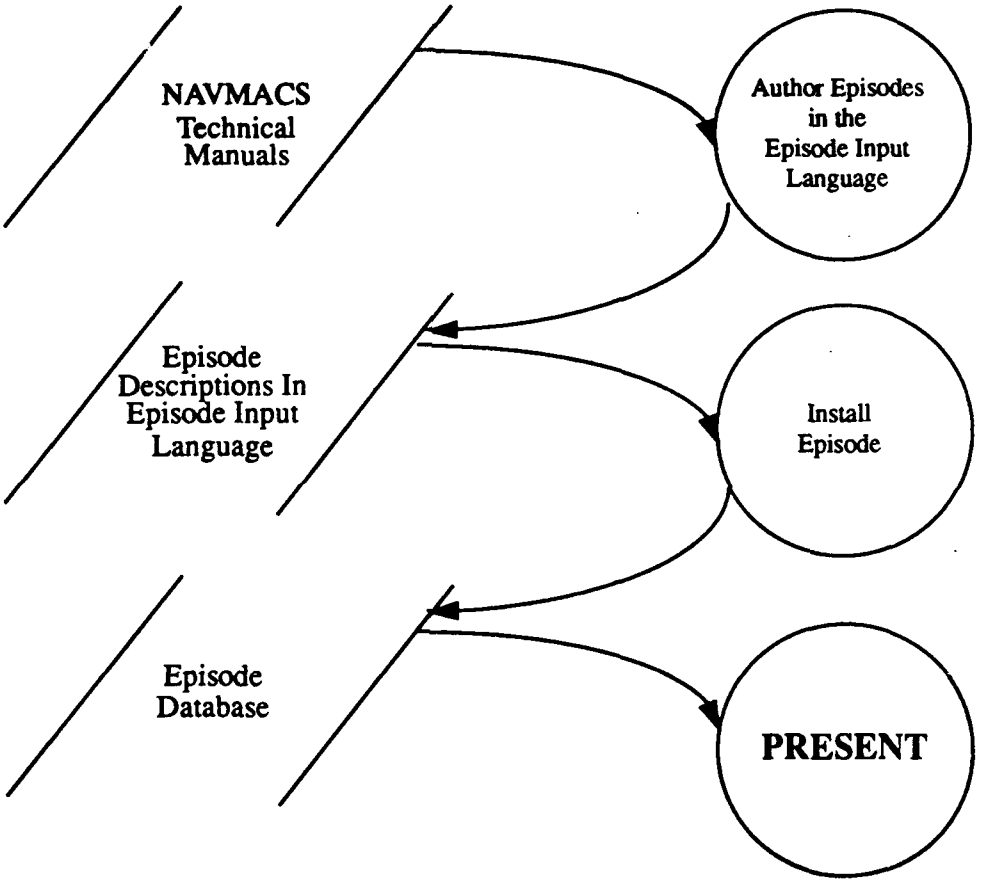


Figure A-4. NAVMACS Episode Authoring System.

The NAVMACS TAE software is driven from an episode database. This database contains information about each episode, including the faulty piece of equipment, and about the states of other components in the system. Each episode represents a snapshot of a faulty system.

An episode is authored directly into a structured format, or language. The language is referred to as the TAE Episode Input Language (EIL). The definition of the EIL provides a foundation for specifying episodes for the NAVMACS hardware. To author a troubleshooting

episode, the lesson developer uses a raw text editor (vi or Wordstar in non-document mode) to write an episode in the EIL, then installs the episode into the episode database.

To process the EIL files and maintain the episode database, the following programs (authoring tools) have been created:

1. Episode Database Creation Program (CREATEDB) - Creates the episode and student history databases. The databases must be created before any episodes or student histories can be added to the databases.
2. Episode Installation Program (ADDEP) - Installs an episode into the episode database once the structure of the episode database has been defined. The episode must be written in the EIL
3. Episode Deletion Program (DELEP) - Deletes an episode from the episode database.

Episode Input Language (EIL) Rules

The EIL syntax and semantic rules are as follows:

1. An episode is divided into three sections: Episode Overview, System-Level Information, and Equipment-Specific information.

   The episode overview is mandatory; system-level information and equipment-specific information are optional. The order indicated above must be respected: the episode overview is placed before the system-level information; the system-level information is placed before the equipment-specific information. For a section that is optional, the defined defaults will be used when that section is not specified.

2. The EIL is built from a set of keywords.

   Keywords are case-insensitive: They can be in uppercase or lowercase (this document will always print keywords in uppercase). Keywords are used to begin sections and commands.

3. The keywords beginning the three sections are OVERVIEW, SYSTEM-LEVEL, and EQUIPMENT-SPECIFIC.

4. Anything after a semicolon is a comment; a semicolon can appear anywhere on a line.

   Here is a typical outline of an episode definition:

```
;  This is the first line of the episode definition.
;  Comments appear after semicolons.
;
;  The first section in an episode definition is the overview. The overview is signaled by the
;  OVERVIEW keyword.
;
OVERVIEW
;
;  Episode overview information would be placed here!!
;  See below for the contents of the overview section.
;
```

```
;   The next section is system-level information, signaled by the SYSTEM-LEVEL
;   keyword.
;
SYSTEM-LEVEL
;
;   System-level information is placed here.
;
;   The last section is equipment-specific information, signaled by the keyword
;   EQUIPMENT-SPECIFIC.
;
EQUIPMENT-SPECIFIC
;
;   Equipment-specific information is placed here.
;
;   End of episode definition!
```

5. Data is assigned to a keyword with the equal sign (=).

6. Aside from keywords, comments, and =, the only other tokens that can occur in an episode definition must represent data. When a datum is a character string, it must be enclosed in quotes.

EIL Keyword Definitions

The definitions of the EIL keywords are presented in Tables A-1, A-2, and A-3. Table A-1 defines the keywords in the episode overview section. Table A-2 defines the keywords in the system-level section. Table A-3 defines the keywords in the equipment-specific section.

Table A-1

Definition of Keywords in the Episode Overview Section

Keyword	Description
OVERVIEW	Identifies the beginning of the episode overview section. Mandatory. Must be first keyword in an episode file. No value is assigned to this keyword.
FAULTED-EQUIP-NAME	Identifies the faulted equipment for the episode. Mandatory. Must be assigned one of the following value keywords ANUSH26V, ANUSQ69V, ANUYK20V, CV3333U, ON143VUSQ, RD397U,or TT624V5UG. TSECKG36 keyword is not allowed.
EPISODE-NUMBER	The episode's identification number. Mandatory. Must be an integer between 1 and 10.
FAULTED-RD	The episode's faulted reference designator. Mandatory. The rd should be enclosed in quotes.
GOOD-FAULTS	An integer identifying the number of GOOD-FAULTS. If there are no GOOD-FAULTS, this should be assigned NONE. If there are GOOD-FAULTS, the subsequent lines will hold the GOOD-FAULT rds, one rd per line. The rds should be enclosed quotes. GOOD-FAULTS is optional; it defaults to NONE.
SYMPTOMS	Either the filename of a graphic picture file OR an integer identifying the number of text lines the symptoms will occupy. This number must be between one and five. The symptom lines should follow on subsequent lines.
PRESENTATION-MODE	Presentation mode for the episode Optional (default is EVALUATION). If this keyword is present, it must be assigned one of the following value keywords: DIRECTIVE, GUIDED, INSTRUCTOR-HALT, or EVALUATION.
MIN-NUM-TESTS	Identifies the number of proof points for the episode (number of lone proof points plus the number of proof point pools). Optional (defaults to zero). If this keyword is present, it must be assigned a value greater than or equal to zero.

Table A-2

Definition of Keywords in the System-Level Section

Keyword	Description
SYSTEM-LEVEL	Identifies the beginning of the system-level section. No value is assigned to this keyword.
EQUIP-EVALS	Identifies the beginning of the assignment of evaluations to each piece of equipment. No value is assigned to this keyword. The lines following this keyword should be the evaluation assignments. The default is for the FAULTED-EQUIP-NAME to be GOOD and each other piece of equipment to be OUT-OF-BOUNDS. Legal evaluations are: GOOD, ILLOGICAL, or OUT-OF-BOUNDS.
FRONT-PANELS	Identifies the beginning of the assignment of picture filenames to each piece of equipment. No value is assigned to this keyword. The lines following this keyword should be the filename assignments. Filenames must be enclosed in quotes. The defaults for each piece of equipment are as follows: "ush26.pic", "usq69.pic", "uyk20.pic", "cv3333.pic", on143.pic", "rd397.pic", "kg36.pic", and "tt624.pic".
ANUSH26V ANUSQ69V ANUYK20V CV3333U ON143VUSQ RD397 TSECKG36 TT624V5UG	These keywords identify each piece of ANUSQ69V equipment. They are used to assign EQUIP-ANUYK20V EVALS and FRONT-PANELS.
FALLBACK	This keyword identifies the beginning of the section of tests known as "fallback tests". No value is assigned to this keyword.

Table A-2 (Continued

Keyword	Description
BROADCAST-D BROADCAST-I PR1-D PR1-I PR2-D PR2-I	These are the various fallback test results. The results are names of text files. If the results are not a text string enclosed in quotes, they should be assigned a value keyword of OUT-OF-BOUNDS, ILLOGICAL, or INVALID. Example: ; FALLBACK BROADCASTD = out-of-bounds BROADCAST-I = "bci2.txt" PR1-D = "pr1d2.txt" PR1-I = "pr1i1.txt" PR2-D = out-of-bounds PR2-I = out-of-bounds ;
LOAD-OPERATIONAL-PROGRAM	This keyword identifies the beginning of the load-operational-program tests. No value is assigned to this keyword.
BROADCAST-D BROADCAST-I PR1-D PR1-I PR2-D PR2-I	These are the various results of loading operational programs. The results are one-line-text results. If any of the results are not a text string enclosed in quotes, they should be assigned a value keyword of OUT-OF-BOUNDS, ILLOGICAL, or INVALID. Example: ; LOAD-OPERATIONA-PROGRAM DRIVE0OP = out-of-bounds DRIVE1OP = "Program would not load" ;

Table A-3

Definition of Keywords in the Equipment-Specific Section

Keyword	Description
EQUIPMENT-SPECIFIC	Identifies the beginning of the equipment-specific section. No value is assigned to this keyword. This section will vary depending upon the FAULTED-EQUIP-NAME of the overview section. EQUIPMENT-SPECIFIC section for each type of NAME is listed below.
	ANUSH26V
MAINT-PANEL	This keyword is assigned the picture filename of the equipment's maintenance panel.
TESTS	Begins the definition of various diagnostic tests. No value is assigned to this keyword.
FIELD-OPERATING-TESTS	Begins the definition of certain diagnostics known as "field operating tests". No value is assigned to this keyword.
DRIVEW0-LOAD DRIVE1-LOAD RD397-LOAD	These are the specific kinds of field-operating-tests. One of two kinds of values are assigned to these keywords: either the keyword "GOOD", indicating that the test does function on the piece of equipment, or "non-operational", indicating that the piece of equipment cannot perform the test. If "non-operational", a text string is assigned to the keyword. This text string is a one-line explanation to the troubleshooter telling that the test cannot be performed (see example below).
FUNCTIONAL MANUAL-INTERVENTION WRITE-COMPATIBILITY READ-COMPATIBILITY TRANSFERABILITY ENDURANCE SERVO-AMPLIFIER DATA-RESOLUTION	If one of the kinds of field-operating tests (described above) is "GOOD", these specific field operating tests can be performed by the troubleshooter. If the result of any of these tests is not enclosed in quotes, the assigned value should be one of the keywords OUT-OF-BOUNDS, INVALID, or ILLOGICAL.

Table A-3 (Continued)

Keyword	Description
	Example: ; TESTS FIELD-OPERATING-TEST DRIVE0-LOAD = "Program will not load." note --> since Drive0-load is non-operational, all tests under it are out of bounds: FUNCTIONAL = out-of-bounds MANUAL-INTERVENTION = out-of-bounds WRITE-COMPATIBILITY = out-of-bounds READ-COMPATIBILITY = out-of-bounds TRANSFERABILITY = out-of-bounds SERVO-AMPLIFIER = out-of-bounds DATA-RESOLUTION = out-of-bounds DRIVE1-LOAD = good FUNCTIONAL = "an example text string" MANUAL-INTERVENTION = out-of-bounds WRITE-COMPATIBILITY = out-of-bounds READ-COMPATIBILITY = out-of-bounds TRANSFERABILITY = out-of-bounds ENDURANCE = out-of-bounds SERVO-AMPLIFIER = out-of-bounds DATA-RESOLUTION = out-of-bounds ;
CONTROL-CIRCUITS-TEST	This keyword begins a section of certain kinds of tests called control-circuits. No value is assigned.
PROTECTION MAINTENANCE	These identify the results of specific control-circuits tests. If the result of any of these tests is not enclosed in quotes, then the assigned value should be an evaluation keyword other than *GOOD* (*OUT-OF-BOUNDS*, *ILLOGICAL*, or INVALID). Example: ; TESTS CONTROL-CIRCUITS-TEST PROTECTION = out-of-bounds MAINTENANCE = "Drive 0 MPC test fails 19"

Table A-3 (Continued)

Keyword	Description
POWER-CIRCUITS-TEST	This identifies the power circuits test result. If the result of this test is not enclosed in quotes, the assigned value should be an evaluation keyword other than GOOD (OUT-OF-BOUNDS, ILLOGICAL, or INVALID). Example: ; TESTS POWER-CIRCUITS-TEST = out-of-bounds ;
STEP-PROCEDURES	Begins the definition of various step procedure tests. No value is assigned to this keyword.
N.N.N	N.N.N is a numerical step category. Eighty-one different step categories are allowed here: 6 . 2, 6 . 2 . 1, 6 . 3, 6 . 3 . 1 ... 6 . 3 .7 6 6 . 4, 6 . 4 .
S . = 1	S is a step number. Allowable values are 1 ...99. The step number is assigned a value of 1 through 5, indicating how many lines of text result follow. Once a category is listed, several step letters may be listed below it. Example: ; STEP-PROCEDURES 6 . 2 1. = "GR1 = 003152" 6 . 3 . 7 1. = 1 "The indicator is not lit" 2. = 1 "Three indicators are now lit" ;
PROOF-PT-STATUS	Proof point status of the specified step-procedure test. If this keyword is not present, the step-procedure is not a proof point.

Table A-3 (Continued)

Keyword	Description
MAINT-PANEL	This keyword is assigned the picture filename of the equipment's maintenance panel.
STEP-PROCEDURES	Begins the definition of various step procedure tests. No value is assigned to this keyword.
	ANUSQ-69V
NN-NN	NN-NN is a numerical step category. Ninety-one different step categories are allowed here: 10-12, 10-13, 10-14 .. 10-96
A.= 1	A is a step letter. Allowable step letters are A ... Z, AA ... AZ. Once a category is listed, several step letters may be listed below it. The step letter is assigned a value of 1 through 5, indicating how many lines of text result follow. Example: ; STEP-PROCEDURES 10-12 C. = 1 "Air movement present." E. = 1 "DC OPERATOR INDICATOR lit" 10-17 N. = 1 "Get correct message" ;
PROOF-PT-STATUS	Proof point status of the specified step-procedure test. If this keyword is not present, the step-procedure is not a proof point.
	ANUYK-20V
MAINT-PANEL	This keyword is assigned the picture filename of the equipment's maintenance panel.
POWER-UP-DIAGNOSTICS	Begins the definition of power-up/diagnostic steps. No value is assigned to this keyword.

Table A-3 (Continued)

Keyword	Description
POWER-ON MICRO PROGRAM-LOADING CP-MEMORY IO OPTIONS LONG-MICRO	These are the different categories for power-up/diagnostic tests. No value is assigned to these keywords. These can be thought of as subsections under POWER-UP-DIAGNOSTICS. Step numbers and step results will follow each of these subsections.
N	N represents a step number. The step number must be a positive integer greater than or equal to zero. An example: ; POWER-UP-DIAGNOSTICS POWER-ON ; 11-17. 8. = "POWER BLOWER indicator lit." ;
PROOF-PT-STATUS	Proof point status of the specified power-up/diagnostic test. If this keyword is not present, the power-up/diagnostic test is not a proof point.
REG-ACCESS	Identifies which power-up/diagnostic, when performed, will give the user register access. This line should be of the form: REG-ACCESS = POWER-ON, 14 A comma must separate the step "subsection" and step number.
REG-ACCESS STATUS	A string representing the registers displayed when the user has gained register access. Must be enclosed in quotes, and must appear on one line.
	CV3333U
SELF-CARD-TESTS	Identifies the beginning of self-card test definitions. No value is assigned to this keyword. The different types of self-card-tests follow. If the result of any of these tests is not enclosed in quotes, the assigned value should be an evaluation keyword other than GOOD (OUT-OF-BOUNDS, ILLOGICAL, or INVALID).

Table A-3 (Continued)

Keyword	Description
VOICE	One line of text enclosed in quotes.
LAMP	One line of text enclosed in quotes.
DIGITAL	Filename of graphic result enclosed in quotes.
ANALOG	One line of text enclosed in quotes.
POWER	One line of text enclosed in quotes.
SELF	Filename of graphic result enclosed in quotes.
PROOF PT-STATUS	Proof point status of the specified self-card-test. If this keyword is not present, the self-card-test is not a proof point.
ON143VUSQ	
REF-DESIGNATORS ADJUST CONTINUITY CURRENT FREQUENCY LOGIC METER VOLTAGE WAVEFORM	Refer to TT624V5UG.
SELF-TESTS	This keyword begins self-test definitions. No value is assigned to this keyword.
LESS-CRYPTO WITH-CRYPTO INDICATION CONTROLLED	These are the legal ON143VUSQ keywords. Each one's result will be one line of text.
RD397U	
MAINT-PANEL	This keyword is assigned the picture filename of the equipment's maintenance panel.
REF-DESIGNATORS	This keyword begins reference designator definitions. No value is assigned to this keyword. After this keyword, a list of reference designators will occur.

Table A-3 (Continued)

Keyword	Description
PROOF-PT-STATUS	Proof point status of the specified rd. If this keyword is not present, the rd is not a proof point.
ADJUST CONTINUITY CURRENT FREQUENCY LOGIC METER VOLTAGE WAVEFORM	These are the different reference designator tests. The value assigned to WAVEFORM should be a picture filename. The value assigned to any of the other tests is one line of text enclosed in quotes. If any of these tests is not listed for an rd, the test is an invalid check.
LOCAL-OPERATIONS	Identifies the beginning of local operations tests. No value is assigned to this keyword. The different types of local-operations follow. If the result of any of these tests is not enclosed in quotes, the assigned value should be an evaluation keyword other than GOOD (OUT-OF-BOUNDS, ILLOGICAL, or INVALID).
SLEW	One line of text enclosed in quotes.
TEST	One line of text enclosed in quotes.
LEADER	One line of text enclosed in quotes.
DUPLICATE	One line of text enclosed in quotes.
REMOTE-OPERATIONS	Identifies the beginning of remote operations tests. No value is assigned to this keyword. The different types of remote-operations follow. If the result of any of these tests is not enclosed in quotes, the assigned value should be an evaluation keyword other than GOOD (OUT-OF-BOUNDS, ILLOGICAL, or INVALID).
REMOTE-READ	One line of text enclosed in quotes.
REMOTE-PUNCH	One line of text enclosed in quotes.
TT624V5UG	
MAINT-PANEL	This keyword is assigned the picture filename of the equipment's maintenance panel.

Table A-3 (Continued)

Keyword	Description
MASTER-CLEAR	Identifies the master clear text result for this episode.
REF-DESIGNATORS	This keyword begins reference designator definitions. No value is assigned to this keyword. After this keyword, a list of reference designators will occur.
PROOF-PT-STATUS	Proof point status of the specified rd. If this keyword is not present, the rd is not a proof point.
ADJUST CONTINUITY CURRENT FREQUENCY LOGIC METER VOLTAGE WAVEFORM	These are the different reference designator tests. The value assigned to WAVEFORM should be a picture filename. The value assigned to any of the other tests is one line of text enclosed in quotes. If any of these tests is not listed for an rd, the test is an invalid check.
SELF-TESTS	This keyword begins self-test definitions. No value is assigned to this keyword.
AL24-DETECT	Filename holding al24/detect results.
AL24-IGNORE	Filename holding al24/ignore results. The standard names for these files will be: "al24d.txt" and "al24i.txt". The structure of al24 text files (i.e., "al24d.txt" and "al24-i.txt") should be as follows: The first five lines that do not begin with a semicolon as the result for test two; ...; the last five lines that do not begin with a semicolon as the result for test 16.
LOOPBACK	Identifies the loopback text-result.

EPISODE PRESENTATION REQUIREMENTS

Overview

This section describes the functional requirements of the NAVMACS TAE Episode Presentation Program (PRESENT). The PRESENT program delivers troubleshooting episodes to Navy electronics students (troubleshooters) and collects their progress in a student history database.

Each troubleshooting episode begins by showing the student the symptoms indicating how the NAVMACS system is malfunctioning. At that point, the troubleshooter chooses simulated tests to perform on the system, and the PRESENT program displays the results that the real equipment would show. The troubleshooting session terminates when the troubleshooter isolates the system fault and simulates replacement of the defective component.

During the presentation process, every meaningful action the troubleshooter makes is time-stamped and recorded in the student history database. The presentation timer begins when the episode symptoms are displayed and terminates when the system fault is replaced. If the troubleshooter tests a proof point, it is noted for credit at a later time in the evaluation stage.

The functional requirements of PRESENT fall into five groups:

1. User-Interaction Requirements - Requirements dictating the appearance and use of the PRESENT program.
2. Administrative Requirements - Functions which allow an instructor to set up specific troubleshooting episodes for a student to run.
3. System Level Requirements - Simulated NAVMACS tests that are available for every piece of equipment in the NAVMACS system.
4. Equipment Level Requirements - Simulated NAVMACS tests that are unique to each piece of equipment in the NAVMACS system.
5. Miscellaneous Requirements - Requirements that do not fall into the above categories.

Each group is described in detail in the following subsections.

User-Interaction Requirements

The PRESENT program's menu hierarchy is illustrated in Figures A-5 and A-6. Figure A-5 provides the PRESENT administrator menu and system level menu. Figure A-6 provides the menu hierarchy for the equipment specific tests.

[PRESENT administrator menu]	
Quit	
[Test episodes]	
ENTER PASSWORD	
** ENTER TROUBLESHOOTER'S SSN **	
[Test episodes menu]	Instructor selects which
[AN/USH-26(V) Episodes]	piece of equipment the
Episode #1 - Episode#10	episode will come from;
AN/USQ-69(V) Episodes]	and the episode number (1
Episode #1 - Episode#10	through 10).
[AN/UYK-20(V) Episodes]	
Episode #1 - Episode#10	
[CV-3333/U Episodes]	
Episode #1 - Episode#10	
[ON-143(V)/USQ Episodes]	
Episode #1 - Episode#10	
[TSEC/KG-36 menu]	Has no episodes
[RD-397/U Episodes]	
Episode #1 - Episode#10	
[TT-624(V)5/UG Episodes]	
Episode #1 - Episode#10	
** DISPLAY EPISODE SYMPTOMS **	Episode symptoms delivered to troubleshooter
[System level menu]	Troubleshooting begins
Review symptoms	
[Front panels]	
AN/USH-26(V) - TT-624(V)5/UG	
[Fallback]	
Broadcast / parity detect	
Broadcast / parity ignore	
UGC20 to PR1 / parity detect	
UGC20 to PR1 / parity ignore	
UGC20 to PR2 / parity detect	
UGC20 to PR2 / parity ignore	
[Load operational program]	
USH-26 Drive 0	
USH-26 Drive 1	
RD-397	
[Equipment-specific tests]	See next two pages
Replace LRU	
Instructor exit	

Figure A-5. PRESENT Menu Hierarchy for Main Menu and System Level Menu.

Equipment-specific tests]
- [AN/USH-26(V) menu]
  - Maintenance panel
  - [Self tests]
    - [Field operating (Drive 0)]
    - [Field operating (Drive 1)]
    - [Field operating (RD397)]
      - Functional
      - Manual intervention
      - Write compatibility
      - Read compatibility
      - Transferability
      - Endurance
      - Servo amplifier
      - Data resolution
    - [Control circuits]
      - Protect
      - Maintenance
    - Power circuits
  - [Step procedures]
    - Select step category
    - Select step number
    - Perform step
    - Return
  - Replace LRU
- [AN/USQ-69(V) menu]
  - Maintenance panel
  - [Step procedures]
    - Select step category
    - Select step number
    - Perform step
    - Return
  - Replace LRU
- [AN/UYK-20(V) menu]
  - Maintenance panel
  - [Power-up/diagnostics]
    - Power-on
    - Micro
    - Program loading
    - Cp/memory
    - I/O
    - Options
    - Long micro
    - Micro end
    - Loop
  - Register access
  - Replace LRU

This is a sub-menu to all three Field Operating Tests: Drive 0, Drive 1, and RD397

Figure A-6. PRESENT Menu Hierarchy for Equipment-Specific Tests.

[CV-3333/U menu]
    [Self card test]
        Voice
        Lamp
        Digital
        Analog
        Power
        Self
    Replace LRU
[ON-143(V)/USQ menu]
    [Reference designator tests]
        Select reference designator
        [Reference designator test types]
            Adjustment
            Continuity
            Frequency
            Current
            Logic
            Read meter
            Voltage
            Waveform
        Perform
        Return
    [IG self tests]
        Less crypto
        With crypto
        Indicator check
        UYK20 controlled
    Replace LRU
[RD-397/U menu]
    [Reference designator tests]
        -- See ON143 Above --
    [Local operations]
        Slew
        Test
        Leader
        Duplicate
    [Remote operations]
        Remote read
        Remote punch
    Replace LRU
[TSEC/KG-36 menu] -- No Access
[TT-624(V)5/UG menu]
    [Reference designator tests]
        -- See ON143 Above --
    Maintenance panel
    Master clear
    Interlock / master clear
    [Self test menu]
        AI24-parity to detect
        AI24-parity to ignore
    Replace LRU

Figure A-6. (Continued)

Administrative Requirements

1. PRESENT begins with a display of the main menu. Figure A-5 displays the options on the main (administrator) menu. The Quit option returns the user to the operating system.

2. When the Test Episodes option is selected from the main menu, the user (instructor) will be asked to enter a password. An incorrect password will cause the program to return to the main menu.

3. If the password is entered correctly, the instructor will be prompted for a student's social security number. After the social security number has been entered, the instructor will be allowed to select the episode to be presented. Using menus, the instructor will first select which piece of equipment the episode will be drawn from and then will select the particular episode to be presented.

4. If the episode selected has already been run and successfully solved by the student, the instructor will be asked if the episode is to be re-run.

5. If the episode selected has already been started and the student either took an instructor exit or had the episode left open because of a computer failure, the instructor will be asked if the episode is to be continued from where the student left off, or rerun from the beginning.

6. Each piece of equipment, except the TSEC/KG-36, will have 10 episodes. No episodes will be developed for the TSEC/KG-36.

7. Once selected, the presentation software will load the episode from the episode database. The troubleshooting session begins by presenting the episode's symptoms to the troubleshooter.

8. Episode symptoms may be text from one to five lines long or full-page graphic images. If the symptoms are given as text, they must be enclosed in a box that is centered on the screen.

9. After the troubleshooter reads the episode symptoms, PRESENT will display the System Level Menu (described in the next subsection).

10. When a troubleshooting session ends, the instructor must be given the opportunity to keep testing on the same student, or to begin testing on a different student, or to exit. This choice must be protected by having to enter the password again so that students cannot set up the next episode without an instructor.

11. Choosing to test on the same student will eliminate the need to reenter that student's social security number. The instructor will go right into the episode selection menus.

12. Choosing to test on a new student will cause the program to ask for a new social security number before going straight to the episode selection menus.

System Level Requirements

1. Figure A-5 displays the options on the System Level Menu. The Review Symptoms option redisplays the episode's symptoms on the screen. After the troubleshooter reviews the symptoms, they will be cleared from the screen.

2. The Front Panels option allows the troubleshooter to view the front panel of any of the eight pieces of equipment. Front panels will be simulated by Dr. Halo graphic images. Front panels may be out of bounds for an episode. (Note: Dr. Halo is a registered trademark of Media Cybernetics, Inc.)

3. The Fallback option allows the troubleshooter to perform one of six "fallback" tests. Any of the six fallback tests may be out of bounds or illogical for an episode. When in bounds of the episode, fallback test results are 23 lines long.

4. The Load Operational Program option allows the troubleshooter to perform one of three "Load Operational Program" tests. Any of the three load operational program tests may be out of bounds or illogical for an episode. When in bounds of the episode, a load operational program test result is one line long.

5. The Equipment-Specific option allows the troubleshooter to do more testing on a particular piece of equipment, if the equipment is in bounds of the episode. A piece of equipment may be out of bounds or illogical for an episode in which case the troubleshooter will not be allowed to perform tests on that piece of equipment. The next subsection provides more information on the types of tests provided for each piece of equipment.

6. The Replace LRU option allows the troubleshooter to replace a "lowest replaceable unit" in a piece of equipment. Replaceable units are identified by reference designators. When the troubleshooter selects this option, he must specify the reference designator of the component (unit) to be replaced and the piece of equipment the component resides in. If the troubleshooter does not specify the LRU to be replaced before choosing "Perform Replacement" at the menu, no replacement shall be simulated, and no event shall be recorded in the student history. The program should alert the troubleshooter that the LRU is not defined. If the replaced component is the episode's faulted component, the presentation program will inform the troubleshooter of his success and will return to the "Enter Password" form. If the replacement does not alleviate the fault, troubleshooting will continue.

7. The Instructor Exit option allows the instructor to terminate episode presentation. After this option is selected, the instructor must enter a password and a short reason for the termination.

Equipment Level Requirements

1. The Replace LRU option (described in the previous subsection) should be accessible from any of the pieces of equipment. Figure A-6 provides the menu options for the equipment-specific tests.

2. The Maintenance Panel option is available on the AN/USH-26(V), AN/USQ-69(V), AN/UYK-20(V), and the TT-624(V)5/UG. This option allows the troubleshooter to view the maintenance panel of these pieces of equipment. Maintenance panels will be simulated with Dr. Halo graphic images.

3. Reference designator tests are available on the ON-143(V)/USQ, the RD-397/U, and the TT-624(V)5/UG. There are eight types of reference designator tests: adjustment, continuity, frequency, current, logic, read meter, voltage, and waveform. The reference designator test (RD Test) option allows the troubleshooter to perform one of the above identified tests on a specific component in the ON-143(V)/USQ, RD-397/U or in the TT-624(V)5/UG. When the

troubleshooter performs an RD test, he must specify the type of test and the reference designator of the component. The following subsections describe the functional requirements of each piece of equipment.

AN/USH-26(V)

1. The Step Procedures option allows the troubleshooter to perform "Step Procedures" on the AN/USH-26(V). When the troubleshooter selects this option, a submenu will be displayed with these options: Select Step Category, Select Step Number, Perform Step, and Return.
2. The Select Step Category option allows the troubleshooter to select one of the following 81 categories: 6.2, 6.2.1, 6.3, 6.3.1, 6.3.2, 6.3.3, ..., 6.3.76, 6.4, and 6.4.1. The troubleshooter enters the category through an interactive form.
3. The Select Step Number option allows the troubleshooter to select a step number in the range 0 to 99. The troubleshooter enters the number through an interactive form.
4. After selecting a step category and number, the troubleshooter may perform a step procedure with the Perform Step option.
5. Step procedures may be out of bounds or good for an episode.
6. When the troubleshooter performs an out of bounds step procedure, PRESENT must ring the bell and inform the troubleshooter that the step procedure is out of bounds.
7. When the troubleshooter performs a good step procedure, PRESENT will display the result of that step procedure. The result of a AN/USH-26(V) step procedure will be either one-to-five-lines of text or a graphic image.
8. The Diagnostics option will bring up a menu containing AN/USH-26(V) diagnostics: Field Operating (Drive 0), Field Operating (Drive 1), Field Operating (RD-397), Control Circuits, and Power Circuits.
9. The three Field Operating Tests (FOT) can either be operational or not operational for an episode.
10. When the troubleshooter selects a FOT that is not operational, PRESENT must display a single-line message identifying why the FOT is not operational.
11. When the troubleshooter selects a FOT that is operational, PRESENT will display the FOT diagnostics menu, containing the options Functional through Data Resolution. The troubleshooter can then perform a FOT diagnostic. FOT diagnostics may be either out-of-bounds or good.
12. When the FOT diagnostic test is out of bounds, PRESENT must ring the bell and display an out of bounds message.
13. When the FOT diagnostic test is good, the result is one line of text.
14. When the troubleshooter selects the Control Circuits diagnostics option, PRESENT should display a menu with the options Protect and Maintenance. Both of these tests can either be good or out of bounds. When good, the result is a single line of text.

15. The Power Circuits test is a stand-alone diagnostic that can either be good or out of bounds. When good, the result is one line of text.

AN/USQ-69(V)

1. The Step Procedures option allows the troubleshooter to perform "Step Procedures" on the AN/USQ-69(V). When the troubleshooter selects this option, a submenu will be displayed with the options: Select Step Category, Select Step Letter, Perform Step, and Return.

2. The Select Step Category option allows the troubleshooter to select one of the following 84 categories: 10-12, 10-13, 10-14, ..., 10-96. The troubleshooter enters the category through an interactive form.

3. The Select Step Letter option allows the troubleshooter to select a step letter. Step letters for the AN/USQ-69(V) are identified by one or two alphabetical characters; the lettering adheres to the following sequence: A, B, C, ..., Z, AA, AB, AC, ..., AY, AZ. There are no double character values past AZ. The troubleshooter enters the letter(s) through an interactive form.

4. Some step procedure categories do not have "sub" step letters; just entering the category is sufficient for performing the step procedure.

5. After selecting a step category and letter (optional), the troubleshooter may perform a step procedure with the Perform Step option.

6. A step procedure is either out of bounds or good for an episode.

7. When the troubleshooter performs an out of bounds step procedure, PRESENT must ring the bell and inform the troubleshooter that the procedure is out of bounds.

8. When the troubleshooter performs a good step procedure, PRESENT will display the result of that step procedure. The result of a AN/USQ-69(V) step procedure will be either a one-to-five-line text result or a graphic image.

AN/UYK-20(V)

1. The Power-Up/Diagnostics option allows the troubleshooter to perform "Step Procedures" on the AN/UYK-20(V). The troubleshooter will select the category of step procedure to be performed (Power-On through Loop). The troubleshooter must then enter a positive integer identifying the particular step procedure to be performed. At that time, the presentation software will display the result of the step procedure. An AN/UYK-20(V) step procedure result is one line of text.

2. The Register Access option allows the troubleshooter to view the AN/UYK-20(V) registers. To view these registers, the troubleshooter must have first gained "register access". The troubleshooter gains register access by performing a specified step procedure. The "register display" will be represented by one line of text.

CV-3333/U

1. The Self Card Test option allows the troubleshooter to perform CV-3333/U self card tests (Voice through Self). Any of the self card tests can be out of bounds for the episode. "Digital"

and "Self" self card test results, when in bounds of the episode, will be simulated with Dr. Halo graphic images. All other types of self card test results will be one line of text.

ON-143(V)/USQ

1. The RD Test option allows the troubleshooter to perform RD tests as described above.
2. The ON-143(V)/USQ equipment will have two troubleshooting shortcut options for performing RD tests. The student may choose "Red Card Tests" or "Black Card Tests", which will prevent the student from having to enter two common reference designator prefixes for this equipment. The student will be asked to enter "Pin Numbers", which will be appended to the red or black reference designator prefixes. When the student selects "Perform" for a red or black card pin test, the test will be simulated just as though the student selected "Reference Designator Test" and entered the whole reference designator via the keyboard.
3. The IG Self Test option allows the troubleshooter to perform ON-143(V)/USQ self tests (Less Crypto through UYK20 Controlled). Any of the self tests can be out of bounds for the episode. All ON-143(V)/USQ self test results will be one line of text. If the replaced component is the episode's faulted component, the presentation program will inform the troubleshooter of his success and will return to the "Enter Password" form. If the replacement does not alleviate the fault, troubleshooting will continue.

RD-397/U

1. The RD Test option allows the troubleshooter to perform RD tests as described above.
2. The Local Operations option allows the troubleshooter to perform RD-397/U Local operations (Slew through Duplicate). Local operation test results are one line of text.
3. The Remote Operations option allows the troubleshooter to perform RD-397/U Remote operations (Remote read and Remote punch). Remote operation test results are one line of text.

TT-624(V)5/UG

1. The RD Test option allows the troubleshooter to perform RD tests as described above.
2. The Master Clear and Interlock/Master Clear options both produce a one-line text result
3. The Self Test option allows the troubleshooter to perform TT-624(V)5/UG AL24 tests (with the parity switch set to either Detect or Ignore). Both types of AL24 tests (Detect and Ignore) require the troubleshooter to enter an AL24 test number. This number must be between 1 and 16. After the test number has been specified, the presentation software will display the AL24 test result. An AL24 test result is five lines of text.

Miscellaneous Requirements

1. In general, every type of text result should be centered and enclosed in a box. Some text results are eighty (80) characters wide and fill the entire width of the screen; these will not be displayed in a box. Typically these are simulations of printer results.

2. Fallback and TT-624(V)5/UG AL24 test results can contain heart, diamond, and "black mark" characters. AN/USQ-69(V) step procedure results can contain "3 horizontal lines" and "lower right corner" characters. These characters will be represented by characters from the IBM and IBM extended character sets. The heart will be represented by character number three; the diamond will be represented by character number four; the "black mark" will be represented by character number 178; the "3 horizontal lines" will be represented by character number 240; and the "lower right corner" will be represented by character number 217.

3. Since there are no keys on the IBM keyboard for these characters, the following mappings will be used in the episode definition files: a vertical bar ('|', decimal 124) will map into a heart; a caret ('^', decimal 94) will map into a diamond; a backwards apostrophe ('`', decimal 96) will map into a "black mark"; an open brace ('{', decimal 123) will map into "3 horizontal lines"; and a close brace ('}', decimal 125) will map into "lower right corner". Hence, when displaying Fallback, TT-624(V)5/UG AL24 test results, or AN/USQ-69(V) step results, the presentation software must display a heart, diamond, "black mark", "3 horizontal line", or "lower right corner" every time it encounters a vertical bar, caret, backwards apostrophe, open brace, or close brace, respectively.

4. The preceding requirement implies that Fallback, TT-624(V)5/UG AL24 test results, and AN/USQ-69(V) step results may not contain vertical bars, caret, backwards apostrophes, open braces, or close braces.

5. Each episode has one fault; the troubleshooter's goal is to find that fault and replace it. If the troubleshooter replaces a component that is not the fault, it is considered an incorrect solution. During the evaluation portion of TAE, points are deducted from a troubleshooter's score for incorrect solutions. The presentation software must also provide "Good Fault Replacements". These are replacements that do not alleviate the fault, but will not deduct any points from the troubleshooter's score. There will be a maximum of 15 reference designators that can be "Good Fault Replacements" per episode.

6. A troubleshooter may run an episode more than once, but PRESENT only has to keep a student history for the latest session. To help instructors track which episodes have been completed, PRESENT must maintain an "Episodes Completed Record" for each troubleshooter. The Episodes Completed Record identifies which episodes a troubleshooter has finished, which episodes the troubleshooter took an instructor exit on, and which episodes the troubleshooter has not attempted. The Episodes Completed Record is provided during episode selection, as illustrated in Figure A-7.

7. If the instructor selects an episode that has already been presented, a message should be printed warning the instructor that the old student history will be overwritten. At that point, the instructor must verify continuation.

8. A maximum of 255 events will be allowed per troubleshooting episode.

9. The PRESENT program must remain running when a user hits any of the usual MS-DOS keyboard sequences that normally terminate program execution, namely, Control-C, Control-Alt-Delete, Control-S, or Control-Break.

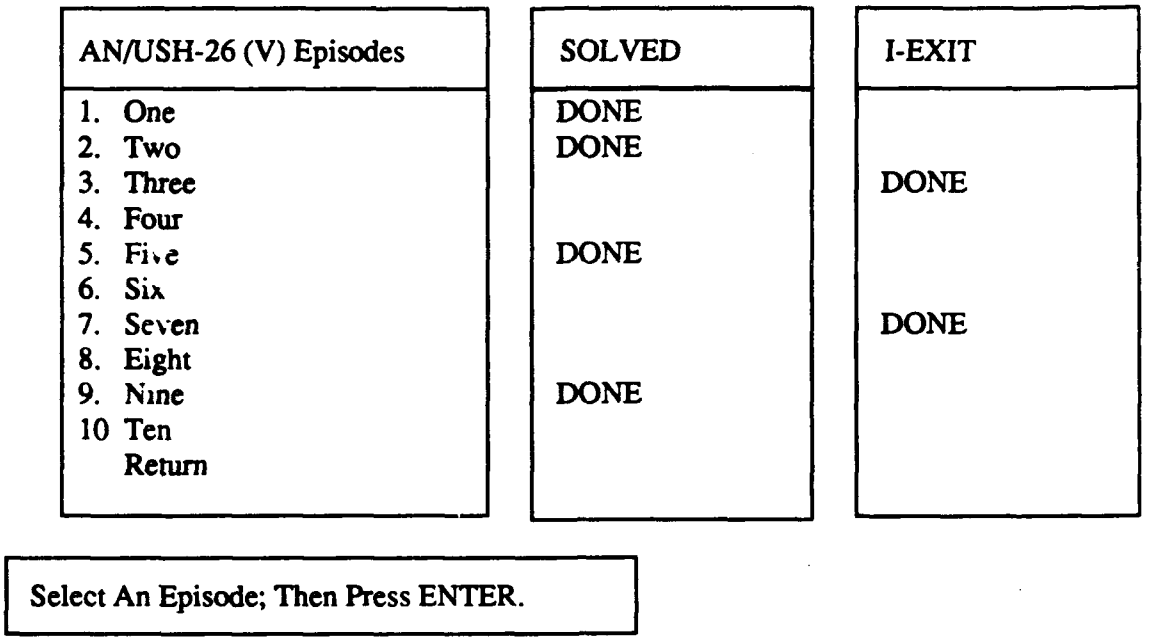


Figure A-7. Episodes Completed Record.

10. Whenever the PRESENT program is expecting the user to enter an "arrow key", as during menu selection, the program must prevent the user from engaging the NumLock key on the keyboard, since engaging the NumLock key prevents the user from using the arrow keys.

STUDENT HISTORY VIEWING REQUIREMENTS

Overview

This section describes the functional requirements of the NAVMACS TAE Student History Viewing Program (EDITVIEW). The EDITVIEW program must provide a variety of tools for viewing and manipulating the student histories recorded by the NAVMACS TAE PRESENT program. Navy instructors and NPRDC personnel will use the EDITVIEW program to monitor the progress of Navy troubleshooters.

Troubleshooting actions are recorded during episode presentation and are stored as events in a student history database. Table A-4 lists the event types stored in the student history database. Each event is time-stamped; time resolution is kept to the minute. The first event in a student history is a login event, time-stamped at zero hours and zero minutes. The last event in a student history is a logout event.

The functional requirements of EDITVIEW fall into five groups:

1. User-Interaction Requirements - Requirements dictating the appearance and use of the EDITVIEW program.
2. Single History Requirements - Functions that operate on a single student history.
3. Multiple History Requirements - Functions that operate on multiple histories.
4. Multiple Scoring Requirements - Functions that show all of the final scores for a specified student, class averages on each episode, class size and class average over all the episodes.
5. Management Requirements - Functions that help manage and organize student histories.

Each group of requirements is described in detail in the following subsections.

User-Interaction Requirements

The hierarchy for the EDITVIEW main menu is illustrated in Figure A-8. The administrative menu shown at the bottom of Figure A-8 is a hidden option that allows the EDITVIEW operator to perform a variety of functions that are protected by password. The administrative menu and its options are described as part of the management requirements at the end of this section.

Table A-4

TAE Event Types in the Student History Database

Event Type	Description	Contents
Diagnostic Test	Operator performed an equipment level test that that cannot be categorized into any other event type	Test Name Evaluation Test Number (if needed) Proof Point Status
Equipment Selection	Operator selected a piece of equipment for further testing	Equipment Evaluation
Fallback	Operator performed a fallback test	Test Name Parity Setting Evaluation
Front Panel	Operator viewed a piece of equipment's front panel	Equipment Evaluation
Instructor Exit	Operator aborted presentation session	Reason
Maintenance Panel	Operator viewed a piece of equipment's maintenance panel	
Load Program	Operator performed a load operational program test	Test Name Evaluation
Login	Signals start of presentation session	
Logout	Signals end of presentation session	
Reference Designator Test	Operator performed a reference designator test	Reference Designator Test Type Evaluation Proof Point Status
Replace LRU	Operator replaced a Lowest Replaceable Unit in a piece of equipment	Equipment Reference Designator Evaluation
Review Symptom	Operator reviewed the episode's symptoms	
Step Procedure	Operator performed a ush26 or usq69 step procedure	Step Category Step Number Evaluation Proof Point Status

[EDITVIEW main menu]
[Examine Individual History]
** ENTER TROUBLESHOOTER'S SSN **
** SELECT EPISODE **
[Display (on screen)]
[Sequence of events]
View next page
View previous page
Scored summary
[Print (on printer)]
Both
Sequence of events
Scored summary
[Edit student data]
** ENTER PASSWORD **
[Add an event]
[Delete an event]
[Change an event]
[Change SSN]
*** ENTER NEW SSN AND CONFIRMATION **
View next page
View previous page
[Return]
Cancel edits
Save edits
Return to editing
Delete history
** Enter confirmation **
[Return]
Examine History: Same Student
Examine History: New Student
Choose another history (same student)
** SELECT EPISODE **
Return to main menu
[Print multiple histories]
Print all
Key on SSN
Key on episode number
Return
[Show Final Scores]
[View Individual Scores]
** ENTER TROUBLESHOOTER'S SSN**
Display Final Scores
Print Final Scores
Return
[View Class Scores]
Display Class Scores
Print Class Scores
Return
[Administrative menu (This selection is BLANK for security)]

These lead to several other menus which allow user to carry out all editing tasks.

Figure A-8. EDITVIEW Menu Hierarchy.

EDITVIEW must show the instructor all of the social security numbers that are in the current student history database whenever it prompts the instructor to enter a student's social security number. It is expected that usually no more than 12 students will be in a student history database at one time, but allowance must be made to show as many social security numbers as will fit on the video monitor at one time. When the user selects an individual history to operate on, the Episodes Completed Record (previously shown in Figure A-7) must be displayed.

Single History Requirements

EDITVIEW must provide the following functions:

1. Display or print a student history.
2. Display or print a summarized analysis of a student history.
3. Allow a student history to be deleted.
4. Allow data in a student history to be edited.

The requirements for each of those functions are described below.

Displaying and Printing Student Histories

When the EDITVIEW operator requests to display or print a student history, EDITVIEW must generate a sequence of troubleshooting events formatted like the one illustrated in Figure A-9. Each troubleshooting event must be formatted into five columns: sequence number, event type, time-stamp, elapsed time since last event, and status information. A header will precede the sequence of events. The header contains a title, a revision number, a revision reason, column titles, and the troubleshooter's social security number. If the student history is in its original form, the revision number and reason are replaced with the word "Original". If the student history has been edited, the revision number and reason for the edit must be included in the header. Figure A-10 illustrates a header with a revision number and reason.

Each event (other than a proof point event) will generate one display/print line. An event that is a proof point will generate two lines, the second line providing proof point status information. If a student history contains too many events to appear on one screen, EDITVIEW must allow the operator to page forward and backward in the student history.

If a student history contains too many events to be printed on one page, EDITVIEW must automatically generate a form feed at the bottom of the page (so that output is not printed on a page perforation).

SEQUENCE OF EVENTS
[Original]

Seq#	EVENT	TIME	ETIME	STATUS INFORMATION SSN: 123456789
1.	Login	00:00	00:00	TT-624(V)5/UG : Episode #1
2.	Review Symptoms	00:01	00:01	
3.	Front Panel	00:06	00:05	AN/USH-26(V) : OUT OF BOUNDS
4.	Front Panel	00:22	00:16	TT-624(V)5/UG : GOOD
5.	Equip Selection	00:45	00:23	AN/USH-26(V) : OUT OF BOUNDS
6.	Equip Selection	01:02	00:17	TSEC/KG-36 : ILLOGICAL
7.	Equip Selection	01:23	00:21	TT-624(V)5/UG : GOOD
8.	RefDes Test	02:00	00:37	A2A18P40 : WAVEFORM : GOOD
9.	RefDes Test	02:12	00:12	A1A3R36 : CURRENT : GOOD Proof Point Pool # 2
10.	RefDes Test	02:30	00:28	A1A3R36 : VOLTAGE : GOOD Lone Proof Point
11.	RefDes Test	02:34	00:04	A1A3P28 : WAVEFORM : INVALID
12.	RefDes Test	02:42	00:08	A1A3P28 : LOGIC : GOOD Proof Point Pool # 1
13.	Diagnostic	03:10	00:28	AL24-DETECT : #2 : GOOD
14.	Diagnostic	03:40	00:30	AL24-IGNORE : #16 : GOOD Lone Proof Point
15.	Diagnostic	04:44	01:04	MASTER-CLEAR : GOOD Lone Proof Point
16.	Review Symptom	05:12	00:28	
17.	Replace LRU	06:19	01:07	A13 : INCORRECT
18.	Fallback	06:31	00:12	BROADCAST : DETECT : GOOD
19.	Fallback	06:31	00:00	UGC20 to PR1 : DETECT : GOOD
20.	Fallback	06:32	00:01	UGC20 to PR2: IGNORE : GOOD
21.	Load Op Program	06:33	00:01	USH-26 Drive 1 : GOOD
22.	Replace LRU	06:34	00:01	A2A18 : GOOD FAULT REPLACEMENT
23.	Replace LRU	06:36	00:02	A2A19 : CORRECT SOLUTION
24.	Logout	06:52	00:16	

Figure A-9. Sequence of Troubleshooting Events.

SEQUENCE OF EVENTS
[Revision #1; Reason: Add step proc's, I-exit]

Seq#	EVENT	TIME	ETIME	STATUS INFORMATION SSN: 123456789
1.	Login	00:00	00:00	AN/USH-26(V) : Episode #5
2.	Review Symptoms	00:01	00:01	
3.	Front Panel	00:06	00:05	AN/USH-26(V) : GOOD
4.	Front Panel	00:22	00:16	TT-624(V)5/UG : GOOD
5.	Load Op Program	00:22	00:00	RD397 Operation : GOOD
6.	Equip Selection	00:22	00:00	AN/USQ-69(V) : OUT OF BOUNDS
7.	Equip Selection	00:23	00:00	AN/USH-26(V) : GOOD
8.	Maint Panel	00:23	00:00	
9.	Diagnostic	00:24	00:00	Fot Drive0 : NOT OPERATIONAL
10.	Diagnostic	00:24	00:00	Fot Drive1 : NOT OPERATIONAL
11.	Diagnostic	00:25	00:00	Fot RD397 : Functional : GOOD
12.	Diagnostic	00:27	00:01	Power Circuits : OUT OF BOUNDS
13.	Step Procedure	00:28	00:00	6.2 : 1 : GOOD
14.	Step Procedure	00:28	00:00	6.3.7 : 1 : GOOD
15.	Step Procedure	00:29	00:00	6.3.7 : 2 : GOOD
16.	Step Procedure	00:29	00:00	6.3.7 : 3 : GOOD
17.	Step Procedure	00:30	00:00	6.3.7 : 4 : OUT OF BOUNDS
18.	Instructor Exit	00:33	00:02	It is lunch time.
19.	Logout	00:33	00:00	

Figure A-10. Revised Sequence of Events.

Displaying and Printing a Summarized Analysis

When the operator requests to display or print a summarized analysis, EDITVIEW must generate a scored analysis like the example provided in Figure A-11.

SCORING SUMMARY

Description	Number (Data)	Score
TOTAL POINTS:		+410.00
FOUND SOLUTION:	YES	- 0.00
TEST POINTS:	6	- 6.00
OUT OF BOUNDS:	6	- 18.00
VALID CHECK:	6	- 1.50
INVALID CHECKS:	5	- 10.00
REDUNDANT CHECKS:	2	- 4.00
ILLOGICAL APPROACH:	1	- 2.00
INCORRECT SOLUTIONS:	1	- 10.00
PROOF POINTS:	2 OF 5	- 6.00
TOTAL TIME:	16 MINUTES	- 8.00
FINAL SCORE:		344.50

Figure A-11. Example of a Scored Analysis.

Figure A-12 identifies the types of student history events affecting each criterion in the summarized analysis. The values in the Number(Data) column of a scored analysis (see Figure A-11) are calculated as follows:

1. Found Solution - This is a Boolean value. If the student replaces the fault, this value gets "YES"; otherwise this value gets "NO".

2. Test Points - number of valid (good) reference designator tests. A contiguous sequence of tests at one test point (reference designator) is counted as only one test point. Furthermore, it is possible to count a particular test point more than once if the operator returns to the test point after performing tests elsewhere. The following examples should illustrate this:

Example 1 - Contiguous sequence of tests at one test point:

Reference Designator	Test Type
TP1	waveform
TP1	voltage
TP2	voltage
TP2	logic
TP2	continuity

**** This sequence is counted as 2 test points ****

Example 2 - Returning to a test point:

Reference Designator	Test Type
TP1	waveform
TP2	voltage
TP2	logic
TP2	continuity
TP1	voltage

**** This sequence is counted as 3 test points ****

3. Out Of Bounds - Number of times troubleshooter went out of bounds on any kind of test. This is a positive integer.

4. Valid Check - Number of valid (good) tests the troubleshooter made. This is a positive integer.

5. Invalid Check - Number of invalid tests the troubleshooter made. This is a positive integer.

6. Redundant Check - Number of times the troubleshooter performs reference designator tests more than once in a row. This is a positive integer.

	LOGIN	REVIEW SYMP	FRONT PANEL	EQUIP SELEC	MAINT PANEL	DIAGNOSTIC	REF DES TEST	FALLBACK	LOAD OP PGM	REPLACE LRU	LOGOUT	INSTR EXIT	STEP PROCS
Total Points													
Found Solution										XXX XXX			
Test Points							XXX XXX						
Out of Bounds			XXX XXX	XXX XXX		XXX XXX	XXX XXX	XXX XXX	XXX XXX				XXX XXX
Valid Checks							XXX XXX						
Invalid Checks							XXX XXX						
Redundant Checks							XXX XXX						
Illogical Approaches			XXX XXX	XXX XXX		XXX XXX		XXX XXX	XXX XXX				
Incorrect Solutions										XXX XXX			
Proof Points						XXX XXX	XXX XXX						XXX XXX
Total Time	XXX XXX										XXX XXX	XXX XXX	

Figure A-12. Data Dependencies for Scored Analysis.

Example: Event	Status Information	#Redundant Checks
Reference Designator Test	TP1:VOLTAGE:GOOD	0
Reference Designator Test	TP1:VOLTAGE:GOOD	1
Reference Designator Test	TP1:WAVEFORM:GOOD	1
Reference Designator Test	TP1:VOLTAGE:GOOD	1
Reference Designator Test	TP2:VOLTAGE:GOOD	1
Reference Designator Test	TP1:CURRENT:GOOD	1
Reference Designator Test	TP1:CURRENT:GOOD	2

7. Illogical Approach - Number of times troubleshooter made an illogical approach on any kind of test. This is a positive integer.

8. Incorrect Solutions - Number of times the troubleshooter replaced a reference designator that was not the faulted reference designator. This is a positive integer.

9. Proof Points - Number of proof points the operator tested. This is a positive integer.

10. Total Time - Number of minutes from login to logout (or instructor exit). This is a positive integer.

The values in the Score column of a scored analysis (see Figure A-11) are calculated using the values in the Number (Data) column and constants that are retrieved from the scoring maximums and scoring weights files. There is one scoring maximums file used by every TAE episode. There is one scoring weights file for each TAE episode. Table A-5 lists the contents of the scoring maximums file. Table A-6 lists the contents of a scoring weights file.

Table A-5

Scoring Maximums

Description	Value	Mnemonic
Total points possible	Real	k.total.points
Max no solution found penalty	Real	k.no.solution
Max bad replacement penalty	Real	k.incorrect
Max illogical approach penalty	Real	k.illogical
Max invalid tests penalty	Real	k.invalid
Max out of bounds penalty	Real	k.out.of.bounds
Max proof point penalty	Real	k.proof.point
Max test point penalty	Real	k.test.point
Max time penalty	Real	k.time
Max valid test penalty	Real	k.valid.check
Max redundant check penalty	Real	k.red.check

Table A-6

Scoring Weights

Description	Value	Mnemonic
Weighted bad replacement penalty	Real	w.incorrect
Weighted illogical approach penalty	Real	w.illogical
Weighted invalid texts pen	Real	w.invalid
Weighted out of bounds penalty	Real	w.out.of.bounds
Weighted proof point penalty	Real	w.proof.point
Weighted test point penalty	Real	w.test.point
Weighted time penalty	Real	w.time
Weighted valid test penalty	Real	w.valid.check
Weighted redundant check penalty	Real	w.red.check

The final score shown in a scored analysis (see Figure A-11) is computed by subtracting the penalties calculated for each criterion from the total points possible for an episode. The values are real numbers that can have a maximum of three digit before the decimal point and two digits after the decimal point. Real numbers may be negative. The final score must never be less than zero. A student's score must be set to zero if the total points to be subtracted exceed the total points possible for the episode. The method for calculating the penalty values for each criterion is provided below.

1. Total points possible = k.total.points
2. If troubleshooter found fault, found solution penalty = 0; else found solution penalty = k.no.solution.
3. Test point penalty = the lesser of k.test.point and w.test.point multiplied by Test Points.
4. Out of bounds penalty = the lesser of k.out.of.bounds and w.out.of.bounds multiplied by Out Of Bounds.
5. Valid check penalty = the lesser of k.valid.check and w.valid.check multiplied by Valid Check.
6. Invalid check penalty = the lesser of k.invalid.check and w.invalid.check multiplied by Invalid Check.
7. Redundant check penalty = the lesser of k.red.check and w.red.check multiplied by Redundant Check.
8. Illogical penalty = the lesser of k.illogical and w.illogical multiplied by Illogical Approach.
9. Incorrect penalty = the lesser of k.incorrect and w.incorrect multiplied by Incorrect Solutions.
10. Proof point penalty = the lesser of k.proof.point and w.proof.point multiplied by minimum.num.tests minus Proof Points.

11. Time penalty = the lesser of k.time and w.time multiplied by Total Time.

Deleting a Student History

EDITVIEW must allow an instructor to remove a specific episode history for a given student. This action will permanently remove that history from the current student history database. The instructor must be given the opportunity to cancel this selection after choosing it.

Editing Data In a Student History

The editing facility must be menu-driven for compatibility with the rest of EDITVIEW and for ease of use. The EDITVIEW operator must enter a password to receive access to event editing. After the correct password has been entered, the edit facility must display a sequence of troubleshooting events similar to that provided by the display facility. The EDITVIEW operator must then be allowed to:

1. Add an event to a student history.
2. Delete one or more events at a time from a student history.
3. Change an event in a student history.
4. Change the Social Security Number for the student history.
5. Page forward or backward in a student history.
6. Exit from event editing with the option to save the changes just made or abandon them.

A number of event editing rules are immediately apparent:

1. The operator should not be allowed to change or delete the first event (login) in a student history.
2. The operator should not be allowed to change or delete the last event (logout) in a student history.
3. If the history contains any revision events as a result of prior editing, it is illegal to change the revision events.
4. The operator should not be allowed to add an event before the first event.
5. All other events must be fully editable. Any field in any event should be easily changeable.

When the operator is finished making changes to a student history file, the EDITVIEW program must prompt the operator to enter a reason for making the edits. The reason will be typed in by the operator into an interactive form in the center of the monitor screen. The reason must be limited to 23 characters. This reason is then saved by the EDITVIEW program as part of a "Revision Event", which is added to the student history after all other events. The next time this student history is displayed or printed, the revision number and reason are displayed at the top, as previously shown in Figure A-10. Previous revision events are displayed at the bottom of the list of events, so that an instructor may view the editing history associated with the given student event history.

Multiple History Requirements

EDITVIEW must provide a function that prints sets of student histories and summarized analyses at one time. The instructor must be able to specify these sets by episode or by social security number. The multiple printing function must allow the EDITVIEW operator to:

1. Print every student history and scored analysis with one command.
2. Print every student history for a single troubleshooter with one command.
3. Print every student history for a single episode with one command.

Multiple Scoring Requirements

EDITVIEW must provide a function that displays:

1. The set of final scores for each student in the student history database.
2. The number of students in the "class" (current student history database) and the overall average of all episodes completed by all students in the class.

Display Student Final Scores

Figure A-13 is an example of the final scores display for one student. The scores are always calculated and shown for the specific set of 14 episodes shown in the figure. It is assumed that each student will complete all of these episodes during the NAVMACS training course.

Final Scores for Student: 123456789

Equipment Name	Episode Number	Final Score	Class Average	Students who Completed
AN/USH-26(V)	# 1	350.00	396.00	10
AN/USH-26(V)	# 2	350.00	379.75	10
AN/USQ-69(V)	# 2	350.00	366.75	10
AN/USQ-69(V)	# 3	350.00	366.25	10
AN/UYK-20(V)	# 2	350.00	358.50	10
AN/UYK-20(V)	# 3	350.00	428.00	10
CV-3333/U	# 1	350.00	392.00	10
CV-3333/U	# 2	Open, no score	339.00	9
ON-143(V)/USQ	# 2	350.00	339.25	10
ON-143(V)/USQ	# 3	350.00	419.50	10
RD-397/U	# 1	350.00	349.50	10
RD-397/U	# 2	350.00	340.00	10
TT-624(V)5/UG	# 1	350.00	355.50	10
TT-624(V)5/UG	# 3	Not Completed	311.75	9

Student's Average = 300.00 for all 14 episodes

Figure A-13. Final Scores Display for One Student.

The Final Score column will indicate if an episode is not completed or was left "open" because of a computer failure. In both of these cases, a zero is counted as the final score for that episode when the student's average is calculated. In reality, no episodes should ever be left open. They should be re-run once the TAE computer is recovered from the failure which caused the episode to be left open.

The Class Average column shows the average score for all of the students for each episode. Again, a zero is averaged in for any incomplete or open episodes left by any student.

The Students who Completed column shows the number of students in the class who actually performed that episode. This is provided as background information to the Class Average column. If a score in the Class Average column is particularly low, it would be beneficial to know if all students actually performed the episode or not.

Display Class Average

The class average feature must show how many students are currently in the class (the student history database) and display the result of averaging each student's average score That is, the average final score will be computed for each student as it is described above, and then all those averages for all the students in the class will be averaged together. This feature must be menu

driven and the operator must be allowed to display the information on the screen or send it to the printer.

Management Requirements

The management requirements include the functions shown in the EDITVIEW administrative menu presented in Figure A-14. This menu must be a "hidden" option on the EDITVIEW main menu. When the operator selects this option, a password must be entered correctly before the EDITVIEW program may display the administrative menu. In this way, all management functions are protected by password.

[Administrative menu (This selection is BLANK for security)]
  ** ENTER PASSWORD **
  [Author scoring constants]
    Change maximums
    Change weights
  [Select Database Archiving]
    Archive current class
    Retrieve class from archive
  Combine databases
  Transfer database to diskette
  Format for statistic
  Delete Student from Database
  Return to main menu
Exit

Figure A-14. EDITVIEW Administrative Menu.

EDITVIEW must allow the operator to perform the following management functions:

1. Author the scoring weights and maximums used in calculating the summarized analysis.
2. Archive class data.
3. Combine the current database with another database from a floppy diskette.
4. Transfer the current student history database to a floppy diskette.
5. Translate the current student history database into a format that a commercial statistical software package, namely Microstat Version 4.0, may use to perform a more sophisticated analysis of troubleshooters' performances.
6. Delete all of the episodes of a particular student from the student history database and then eliminate that student's social security number from the database.

Each of these requirements is described in the following paragraphs.

Authoring Scoring Constants

The function to author scoring constants must allow the EDITVIEW operator to:

1. Modify the scoring maximums file used for all episodes,
2. Create a scoring weights file for an episode.
3. Modify a scoring weights file for an episode.

The contents of these files were previously shown in Tables A-5 and A-6.

Archiving Student History Databases

The archive function must allow the EDITVIEW operator to:

1. Save the current student history database into an operator-named archive file area.
2. Retrieve an archived student history database by name from a list of all archived student history databases.

The program will prompt the operator to name the archive with menus and interactive forms. The name of any student history database archive will consist of seven characters: a leading "S" if the class location is San Diego, California, or a leading "N" if the class location is Norfolk, Virginia, followed by six digits indicating the class date in the form MMDDYY.

Figure A-15 illustrates the organization of student histories into the class archive.

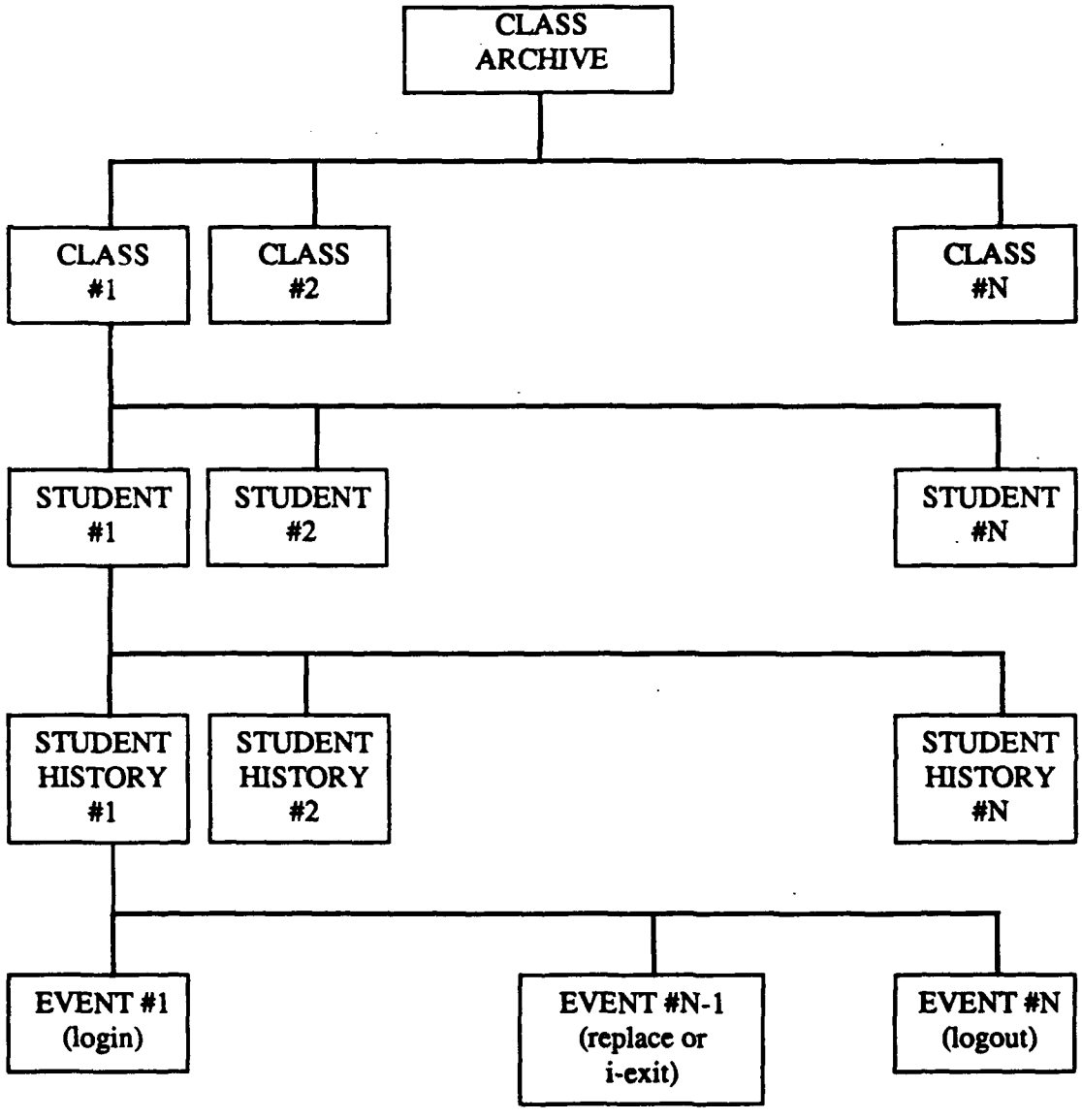


Figure A-15. Student History Organization.

Combine Student History Databases

The function to combine databases must allow the EDITVIEW operator to merge the current student history database with a different student history database from a floppy diskette. This allows the creation of one master database for a class that may have been created from several students working on different computers during their instruction. The resulting database will become the current student history database located in the TAE runtime environment on the computer's hard disk drive.

The function should detect whether any of the students in the two databases have completed the same troubleshooting episode in both databases. If this happens, this facility should report that information to the operator and should NOT attempt to merge (combine) the two databases.

Transfer Student History Database to Diskette

The function to transfer a database to diskette must allow the EDITVIEW operator to transfer the current student history database to a previously formatted double-sided double-density floppy diskette so that the database may be combined with a different NAVMACS TAE student history database.

Format for Statistics

The function to format for statistic analyses must allow the EDITVIEW operator to create a new output file that contains certain student troubleshooting history data in a form that is compatible with the Microstat statistical analysis software package (Version 4.0). The format function will create one Microstat data file from the current student history database. The operator is asked to name this output file, or may use the default name of "STATFILE.MSD".

Microstat data files contain data grouped into "cases". The data in each case are called "variables". Each of these variables may have a unique name of up to 8 ASCII characters associated with it. The variable names for the cases in this file will be simply "V1" through "V673". All the data for a given student will comprise one case. The first variables for each case will be the student's social security number. Following that will be the data for each of the 16 NAVMACS episodes listed below.

1.	AN/USH-26(V)	episode #1
2.	AN/USH-26(V)	episode #2
3.	AN/USQ-69(V	episode #2
4.	AN/USQ-69(V)	episode #3
5.	AN/UYK-20(V)	episode #2
6.	AN/UYK-20(V)	episode #3
7.	CV-3333/U	episode #1
8.	CV-3333/U	episode #2
9.	ON-143(V)/USQ	episode #2
10.	ON-143(V)/USQ	episode #3
11.	RD-397/U	episode #1
12.	RD-397/U	episode #2
13.	TT-624(V)5/UG	episode #1
14.	TT-624(V)5/UG	episode #3
15.	AN/USH-26(V)	episode #4
16.	TT-624(V)5/UG	episode #2

The specific data for each episode and the order of output are listed below.

1. Equipment number (1=AN/USH-26(V) ... 8=TT-624(V)5/UG)
2. Episode number
3. Found Solution (1 = Yes, 0 = No)
4. Number of Test Points
5. Number of Out of Bounds tests
6. Number of Valid Checks
7. Number of Invalid Checks
8. Number of Redundant Checks
9. Number of Proof Points the subject tested
10. Total number of Proof Points in the episode
11. Percentage of Proof Points tested rounded to a whole number
12. Total Time spent on the episode, in minutes
13. Number of Illogical Approaches
14. Number of Equipment Selection Events
15. Number of Front Panel events
16. Number of Maintenance Panel events
17. Number of Fallback Test events
18. Number of Reference Designator test events
19. Number of Replace LRU events
20. Number of Review Symptom events
21. TBD (output a zero)
22. Number of Diagnostic Test events
23. Number of Load Operational Program events
24. Number of Step Procedure events
25. Number of Revision Events
26. Number of Incorrect Replace LRU events
27. Number of Good Fault Replacement Replace LRU events
28. Time until first Reference Designator Test, in minutes
29. Time until first Diagnostic Test, in minutes
30. Total number of steps taken in the episode: Includes Login and Logout, but not Revision events
31. Number of Waveform tests performed
32. Number of Voltage tests performed
33. Number of Read Meter tests performed
34. Number of Logic tests performed
35. Number of Current tests performed
36. Number of Frequency tests performed
37. Number of Continuity tests performed
38. Number of Adjustment tests performed
39. Final Score for the episode
40. TBD (output a zero)
41. TBD (output a zero)
42. TBD (output a zero)

If a student has not completed a certain episode, the output file must contain the Microstat designation for "Missing" data in sufficient quantity to fill up the same number of output bytes normally taken by the episode data.

Delete Student from Database

This function must allow the EDITVIEW operator to completely eliminate one particular student at a time from the current student history database. After the user enters the social security number of the student to be deleted, the program must obtain a double confirmation from the user before actually deleting the student. All of the episodes that student performed, as well as the student's social security number must be deleted from the database.

LOCAL AREA NETWORK REQUIREMENTS

The following requirements must be met:

1. The TAE system must function under a local area network (LAN) made by Corvus Systems and known as Omninet. This particular commercial LAN product was selected by the Navy for this project under a previous work order after evaluating several LANs available for IBM PC compatible computers.

2. The Navy schoolhouse which functions as the test facility for the TAE project has ten Zenith-248 computers in one classroom. One instructor supervises up to ten students at a time, and sets up each TAE troubleshooting episode individually for each student at each separate computer. The TAE system must be made to operate under the Omninet LAN so that the instructor may work at one computer and set up troubleshooting episodes for the students at all of the remote computers. The various computers connected using a LAN are known as "nodes". The instructor's computer is referred to as the "administrator node" and the students' computers as the "student nodes".

3. The TAE system must continue to function normally if the administrator node fails. That is, if the instructor's computer experiences a hardware or software crash, the PRESENT and EDITVIEW programs must still run on the student nodes.

4. The PRESENT and EDITVIEW programs must run in their normal "stand-alone" MS-DOS environment as well as under the Omninet LAN environment.

5. The instructor must be able to access the student history databases on student nodes from the administrator node.

APPENDIX B

TROUBLESHOOTING ASSESSMENT AND ENHANCEMENT (TAE) PROGRAM DESIGN DESCRIPTION

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

This appendix describes the Troubleshooting Assessment and Enhancement Program (TAE) system, designed for the Naval Modular Automated Communications Subsystem (NAVMACS). This document describes the entire TAE system with regard to its software design from the perspective of the software developer. It assumes that the TAE System Requirements Document (SRD) is available for reference (see Appendix A).

The appendix is organized into sections ordered in increasing levels of detail. The first section provides an overview of the system, including a definition of the system context and a brief description of each of the system components. The next two sections present the system-level data flow and the global abstract data dictionary. The remaining sections describe each of the system components in detail, including a data flow diagram, a data dictionary, and implementation conventions.

Some of the terms used throughout the documents are defined below:

Component--The lowest replaceable hardware unit.

Episode--A state of the system to be presented to the student. An episode contains a single fault in a piece of equipment.

Equipment--A piece of hardware that performs a particular function. A piece of equipment consists of one or more components.

Picture--A Dr. Halo picture file to be displayed on the EGA graphics card (640 x 350 pixels, 16 colors).

Proof Point--A test on a component which gives the troubleshooter a particular clue as to the source of the equipment's fault.

Reference designator--Label identifying a physical component in a piece of equipment.

System--A collection of two or more pieces of equipment.

TAE SYSTEM OVERVIEW

The objective of the Troubleshooting Assessment and Enhancement (TAE) Program is to develop a system to assess the proficiency of Navy technicians in troubleshooting a variety of electronic equipment. To satisfy this objective, the TAE system performs the following functions:

1. It allows subject matter experts to author simulated troubleshooting episodes.

2. It presents these episodes to Navy troubleshooters and collects their interactions in a history database.

3. It allows Navy instructors to view, edit, and analyze the collected troubleshooting histories.

Figure B-1 presents a context diagram of the TAE system. The system's major components include software programs (TAE software), microcomputer workstations (TAE hardware), Navy instructors, lessonware authors (subject matter experts), and Navy troubleshooting technicians (troubleshooters).

Each component plays a role in achieving the system's objective. The TAE hardware provides the central processing unit (CPU), graphics screen, disk, memory and input/output (I/O) devices to support the execution of the TAE software. The subject matter experts and Navy instructors generate system lessonware using the episode authoring program. The Navy troubleshooters perform the system lessonware using the student history viewing program.

Custom Software

The following paragraphs describe the custom software components shown in Figure B-1 in more detail.

Episode Authoring Program

The purpose of the episode authoring program is to allow subject matter experts and Navy instructors to author simulated troubleshooting episodes. The name of the episode authoring program is ADDEP (Add Episode).

When developing a troubleshooting episode for equipment, the subject matter expert must specify the equipment's abnormal symptoms (episode symptoms), the faulty component in the equipment causing the abnormal behavior (system fault), and a list of components in the equipment that, when tested, should give the troubleshooter some indication of the system fault (proof points).

Episode Presentation Program

The name of the episode presentation program is PRESENT. The purpose of this program is to present troubleshooting episodes to Navy troubleshooters and to collect their progress in a history database.

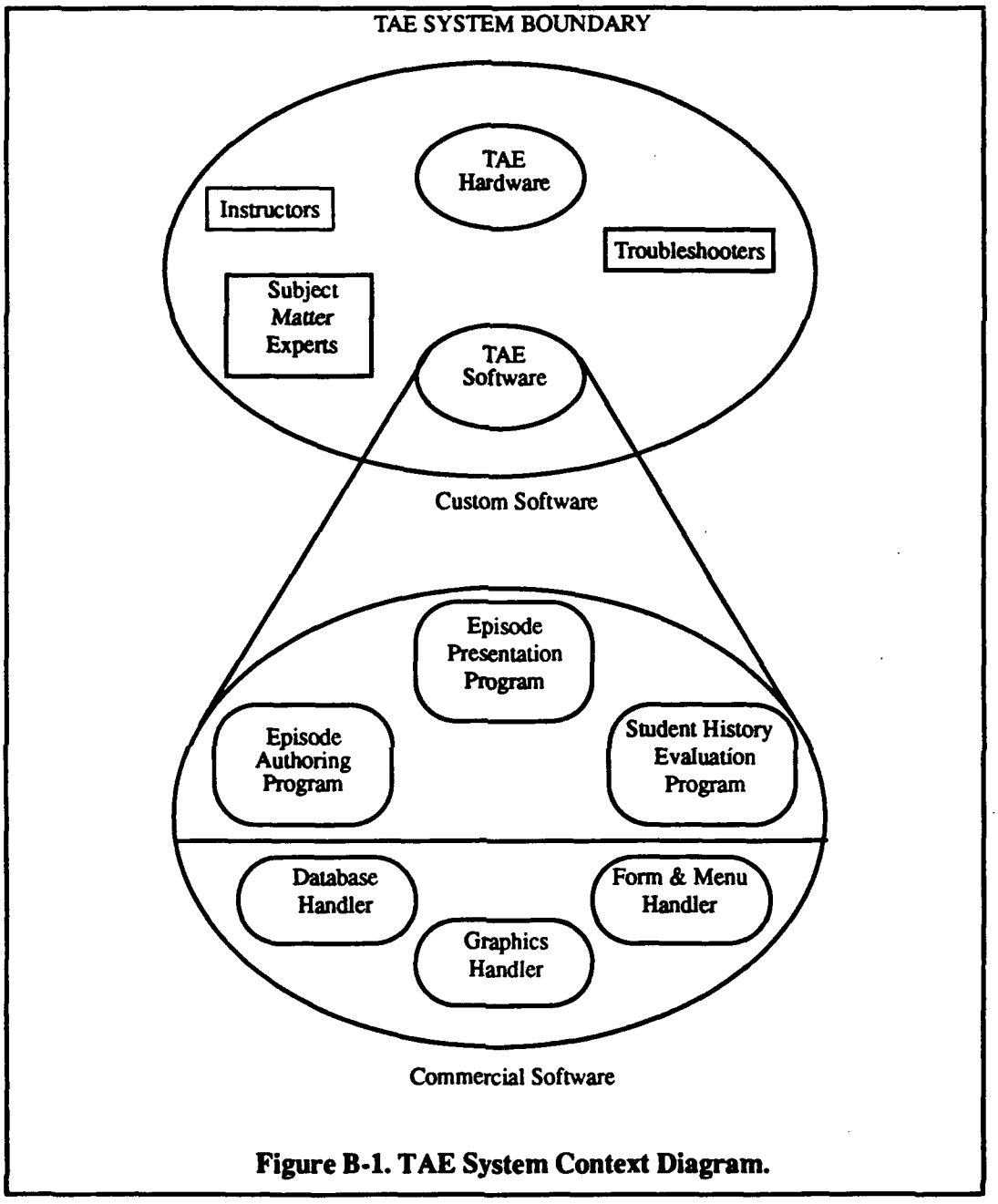


Figure B-1. TAE System Context Diagram.

The episode presentation process begins by delivering the episode symptoms to a Navy troubleshooter. At that point, the troubleshooter indicates tests that he wants to perform, and the PRESENT program displays the results that the real equipment would return. The troubleshooting session terminates when the operator isolates the system fault and uses PRESENT to simulate replacement of the defective component.

During the presentation process, every action the troubleshooter makes is time-stamped and recorded in a student history. The presentation timer begins when the episode symptoms are

displayed and terminates when the system fault is replaced. If the troubleshooter tests a proof point, it is noted for credit at a later time in the evaluation stage.

Student History Viewing Program

The name of the student history viewing program is EDITVIEW. The purpose of this program is to allow Navy instructors to view, edit, and analyze the troubleshooting histories collected by the episode presentation program (PRESENT). Other database management functions are also built into the program.

The "view" function formats and displays each event in the student history in chronological order. The information displayed for each event includes the time (relative to the start time) that the event occurred, the elapsed time since the last event, the event type, and status information related to the event type. Student histories may also be printed on a printer connected to the computer on which EDITVIEW is running.

The "edit" function allows Navy instructors to edit (add, delete, or modify) events in a student history.

The "analyze" function generates a report card summarizing the troubleshooter's performance in the episode. A specialized, statistical software package would be required to do a collective analysis of all troubleshooters' performances.

The other database management functions allow Navy instructors to save (archival) and retrieve student history databases, combine two different student history databases into one larger database, delete a student from the database, and translate the student data into a file that can be read by Microstat, a separate statistical analysis software package.

Commercial Software Utilities

The following paragraphs describe the commercial software components shown in Figure B-1 in more detail. Identification of specific equipment and software is for documentation only and does not imply endorsement.

Form and Menu Handler

The form and menu handler is a screen management utility used by PRESENT and EDITVIEW. Its purpose is to display TAE forms and menus. The TAE code package name of the form and menu handler is "panel_pl".

The form and menu handler is implemented with the commercial software package PANEL Plus, by Roundhill Computer Systems. PANEL Plus provides an interactive screen editor for creating screen images such as forms and menus. If a cosmetic or syntactic change needs to be made to a menu or form, it can be done in a matter of seconds with the panel editor.

PANEL Plus also provides a variety of screen management functions in the form of a linkable software library. This library provides the means of displaying the menus and forms authored with the panel editor.

Graphics Handler

The graphics handler is a graphics display utility used by the PRESENT program. Its primary purpose is to display Dr. Halo full-screen graphics. Its secondary purpose is to provide a set of routines for drawing simple graphics. The TAE code package name of the graphics handler is "graphics".

The graphic drawing routines are implemented with the commercial software package MetaWINDOW/PLUS, by Metagraphics Software Corporation. MetaWINDOW/PLUS provides a wide variety of graphic drawing routines in the form of a linkable software library. Presently, MetaWINDOW/PLUS is used only to draw simple graphics over Dr. Halo images (for example, a box with some status information). (Note. Dr. Halo is a registered trademark of Media Cybernetics, Inc.) If future requirements dictate more advanced graphic capabilities, MetaWINDOW/PLUS is capable of meeting them.

Database Handler

The purpose of the database handler is to provide access to the episode and student history databases. The database handler is used by ADDEP, PRESENT, and EDITVIEW. The package name of the database handler is "tpep_dbs".

The database handler is implemented with the commercial software package c-tree (version 4.1, release F), by FairCom.

TAE SYSTEM DATA FLOW

Figure B-2 illustrates the high-level data flow of the TAE system. Subject matter experts and instructors author episodes using a word processor. The product of a word processing session is an episode stored in an episode definition file. The episode database is built by subject matter experts and instructors using the ADDEP program. ADDEP builds the episode database from episode definition files. These episodes are presented to troubleshooters by the PRESENT program. The student history database is built by the PRESENT program when episodes are delivered to troubleshooters. Student histories may be displayed or modified by the instructor using the EDITVIEW program.

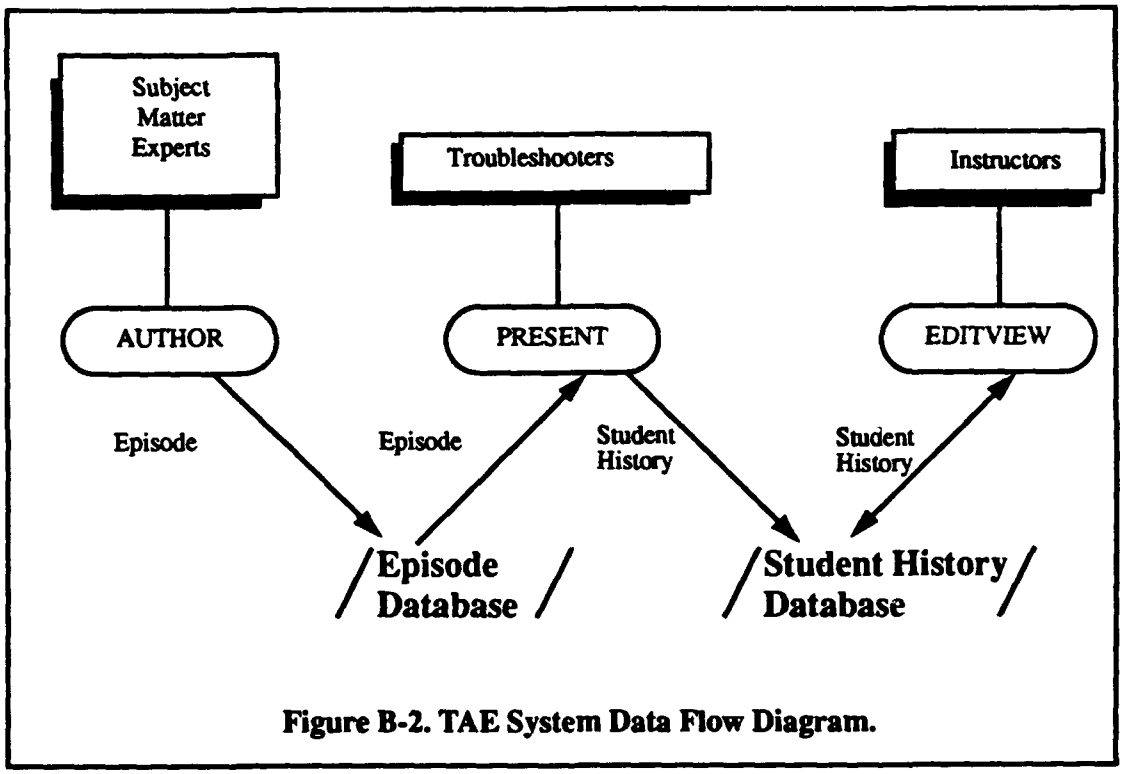


Figure B-2. TAE System Data Flow Diagram.

Figure B-3 illustrates the role that the commercial software utilities play in the TAE system data flow.

The form and menu database is created by the software developer using the PANEL Plus screen editor program. PRESENT and EDITVIEW access the form and menu database through the TAE panel_pl code package.

AUTHOR, PRESENT, and EDITVIEW access the episode and student history databases through the TAE tpep_dbs code package.

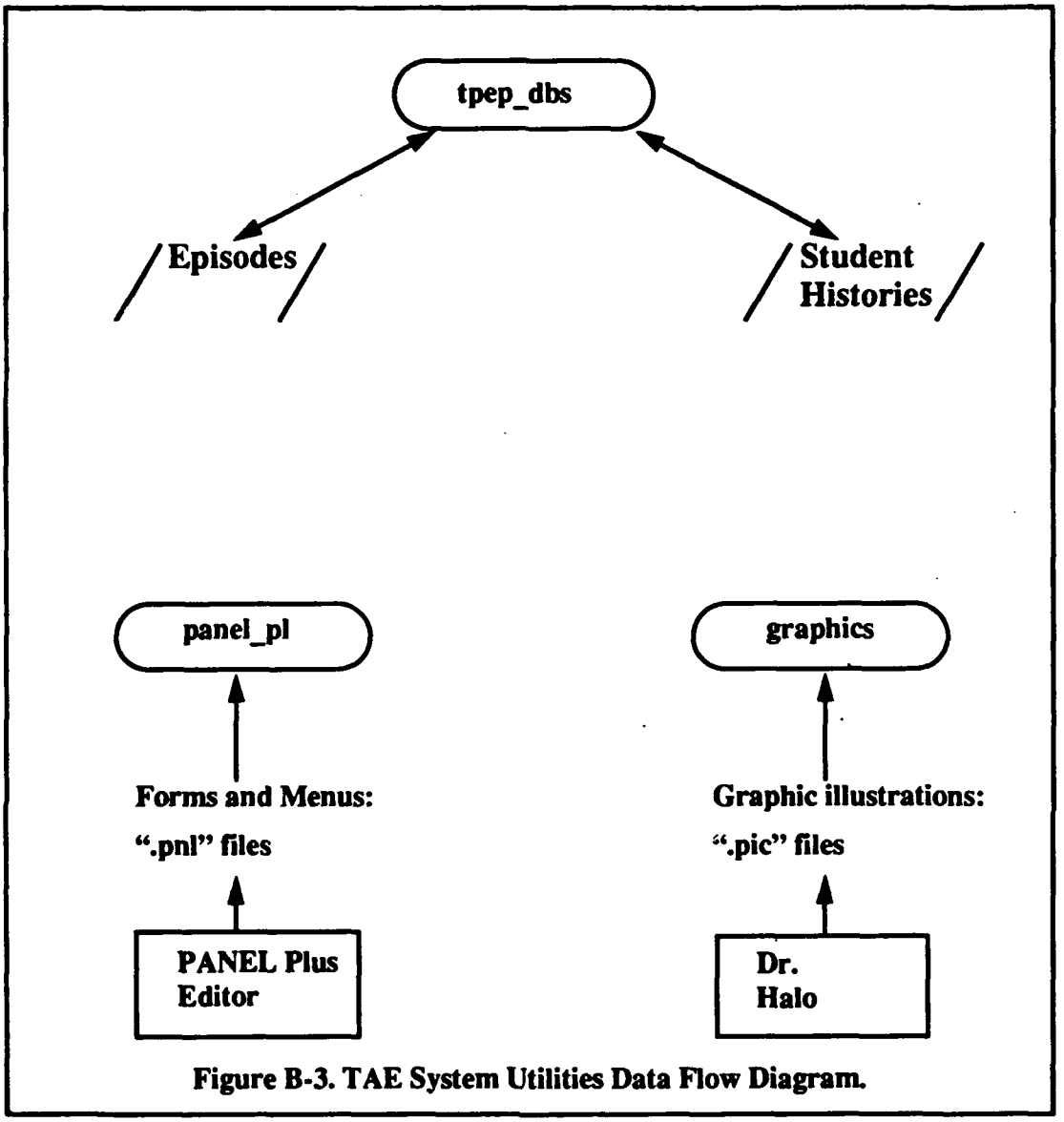


Figure B-3. TAE System Utilities Data Flow Diagram.

The graphics database consists of Dr. Halo picture files with the extension ".pic". Each picture resides in its own file. These pictures are created with the Dr. Halo paint program. PRESENT accesses the graphics database through the graphics utility package.

The following sections define the contents of the episode and student history databases and describe in detail each of the custom and commercial software packages identified in the system data flow diagrams.

TAE Hardware

The host microcomputer for the NAVMACS TAE is a Zenith 248 outfitted with:

1. A 360-kilobyte floppy disk drive
2. A 20-megabyte hard disk
3. 640 kilobytes of RAM
4. A keyboard
5. An Enhanced Graphics Adapter (EGA) card
6. A 13 inch EGA color monitor

Abstract Data Dictionary

The abstract data dictionary presented in Table B-1 defines global data used throughout the TAE system. It also includes NAVMACS related data. Refer to DeMarco (1978) for more information on data dictionary conventions. The numbered notes listed below refer to the numbers in the notes column shown in Table B-1.

1. The name of the subsystem is indicated in Notes column.

2. The episode-key is used as a key into the episode database. It consists of the episode number and the episode's faulted piece of equipment. Given an episode-key, the episode database can be queried for more information about the episode.

3. The NAVMACS system consists of these pieces of equipment. The equipment names here have been stripped of hyphens and parentheses because that is how they are spelled in the software itself when it uses these equipment names as data.

4. Most troubleshooting actions are given one of these evaluations.

5. These are the different events that are stored in a student history.

6. This data item holds an "instructor exit" message. The message should be the reason the instructor has terminated the session prematurely.

7. Min-num-tests identifies the minimum number of tests the troubleshooter should make to be certain of the fault in the system. Min-num-tests is equal to the number of lone proof points plus the number of proof point pools for the episode.

8. Filename of a Dr. Halo graphic image.

9. Proof point status is interpreted as follows:

Proof_pt_status	Meaning
-1	The troubleshooting action is not a proof point.
0	The troubleshooting action is a lone proof point.
1, 2, 3, etc.	The troubleshooting action is a member of the specified proof point pool.

10. This data item holds a NAVMACS reference designator.

Table B-1

NAVMACS Abstract Data Dictionary

Data Name		Description	Notes
a124-test-number	=	integer in range 1 to 16	
a124-result	=	five-line-result	
diagnostic-test	=	[power-on	*AN/UYK-20 (V) *
		micro	
		program-loading	
		cp-memory	
		io	
		options	
		long-micro	
		micro-end	
		loop	
		register-access	
		master-clear	*TT-624 (V) 5/UG*
		interlock-master-clear	
		a124-detect	
		a124-ignore	
		voice	*CV-3333/U*
		lamp	
		digital	
		analog	
		power	
		self	
		less-crypto	*ON-143 (NV) /USQ*
		with-crypto	
		indicator-check	
		uyk20-controlled	
diagnostic-test	=	local-slew	*RD-397 /U*
		local-test	
		local-leader	
		local-duplicate	
		remote-read	
		remote-punch	
		fot-drive0	*AN/USH-26 (V) *
		fot-drive1	
		fot-rd397	
		control-circs-protection	
		control-circs-maintenance	
		power-circuits]	

Table B-1 (Continued)

Data Name	Description	Notes
episode-key	= episode-number + equipment-type	2
episode-number	= positive-integer	
episode-completed-record	= 1 {Boolean} 80	
equipment-type	= [anush26v anusq69v anuyk20v cv3333u on143vusq rd397u tseckg36 tt624v5ug]	3
evaluation	= [out-of-bounds illogical invalid good good-fault-replacement not-operational]	4
event-type	= [login logout equipment-selection front-panel maintenance-panel ref-des-test replace-lru review-symptoms instructor-exit diagnostic-self-test step-procedure]	5
exit-message	= string-30	6
fallback-tests	= [broadcast-parity-to-detect broadcast-parity-to-ignore pr1-parity-to-detect pr1-parity-to-ignore pr2-parity-to-detect pr2-parity-to-ignore]	

Table B-1 (Continued)

Data Name		Description	Notes
faulted-equip-name	=	equipment-type	
field-operating-tests	=	[drive0 drive1 rd397]	
file-name	=	string-14	
five-line-result	=	1 {string-80} 5	
fot-diagnostics	=	[functional manual_intervention write_compatibility read_compatibility transferability endurance servo_amplifier data_resolution]	
full-page-result	=	1 {string-80} 22	
good-fault-replacements	=	1 {rd} MAX-NUM-GOOD-FAULTS	
load-operational-program-tests	=	[drive0 drive1 rd397]	
min-num-tests	=	positive-integer	7
parity-switch	=	[detect ignore]	
picture-result	=	file-name	8
presentation-mode	=	[directive guided instructor-halt evaluation]	

Table B-1 (Continued)

Data Name		Description	Notes
proof-pt-status	=	[not-a-proof-point lone-proof-point	9
		proof-point-pool-one proof-point-pool-two proof-point-pool-three]	
rd	=	string-20	10
rd-test-type	=	[adjust-component continuity current freq logic meter-read volt waveform]	
ssn	=	string-9	
step-category	=	string-8	
symptoms	=	five-line-result picture-result	
text-result	=	string-80	

ADDEP PROGRAM

Description

This program allows Navy instructors to add NAVMACS troubleshooting episodes to the episode database. The ADDEP program takes input files, one at a time, that are created by subject matter experts and uses a two-step process to add them to the episode database. In the first step, ADDEP parses and analyzes the input file to ensure that all of information given fits into the guidelines of the episode input language. If a file "passes inspection" by the first step, the input file is then added to the episode database.

A separate utility called DELEP is provided to allow a subject matter expert to delete an episode from the database. This utility would be used in the case where an episode was added to the database and was later found to be incorrect or inappropriate. DELEP deletes one episode at a time, prompting the user with menus.

ADDEP Abstract Data Dictionary

The data dictionary shown in Table B-2 defines data global to the NAVMACS ADDEP program. The numbered note listed below refers to the number shown in the notes column in Table B-2.

1. Hash-table provides quick lookup of episode input language (EIL) keywords.

Package Dependencies

ADDEP uses the TAE tpep_dbs code package to add episodes to the episode database. DELEP uses the TAE tpep_dbs code package to delete an episode from the database and the TAE panel_pl code package to display menus for user interaction.

Table B-2

ADDEP Abstract Data Dictionary

Data Name		Description		Notes
key	=	string-25		
value	=	integer		
hash-table-entry	=	key	+	
		value		
hash-table		0 {hash-table-entry} 200		1
faulted-rd	=	rd		
good-faults	=	good-fault-replacements		
episode-data-record	=	episode-key	+	
		faulted-rd	+	
		good-faults	+	
		present_mode	+	
		min-number-of-tests		
episode-record	=	episode-data-record		
front-panel	=	file-name		
maintenance-panel	=	file-name		
equipment-data-rec	=	episode-key	+	
		equipment-type	+	
		evaluation	+	
		front-panel	+	
		maintenance-panel		
equipment-records	=	0 {equipment-data-rec} 8		
test-name	=	string-1		
system-level-data-rec	=	episode-key	+	
		test-name	+	
		evaluation	+	
		text-result		
system-level-records	=	0 {system-level-data-rec} 9		
step	=	string-1		
register-access-grant	=	test-name	+	
		step		

PRESENT PROGRAM

Description

This program presents troubleshooting episodes to NAVMACS technicians and collects the technicians' troubleshooting interactions in a student history database. The operator interaction (menu hierarchy) of the NAVMACS PRESENT program is defined in the TAE SRD.

PRESENT implements this hierarchy as a tree data structure, referred to as PRESENT's menu-tree. The main function of PRESENT operates off of this menu-tree, beginning at the root, and then climbing up and down in the menu-tree depending on the operator's menu selections. Refer to PRESENT's abstract data dictionary for listing for PRESENT's main function for more information on the menu-tree driver.

PRESENT Abstract Data Dictionary

The data dictionary shown in Table B-3 defines data global to the NAVMACS PRESENT program. The numbered notes listed below refer to the numbers shown in the notes column in Table B-3. The notes start on p. B-18.

Table B-3

PRESENT Abstract Data Dictionary

Data Name		Description		Notes
key	=	string-25		
value	=	integer		
high-table-entry	=	key value	+ 	
hash-table	=	0 {hash-table-entry} 200		1
current-session-status	=	ssn student-key episode-key sequence-num start-time	+ + + + 	6
form-id	=	[instructor-exit-form a124-step-number-form password-form rd-test-form pin-number-test-form replace-rd-form ssn-form uyk20-step-form ush26-step-form usq69-step-form]		5
menu-id	=	[administrator-menu equipment-menu system-level-menu front-panel-menu fallback-menu load-operational-menu equipment-tests-menu anush26v-tests anusq69v-tests anuyk20v-tests cv3333u-tests on143vusq-tests rd397u-tests tt624v5ug-tests anush26v-diagnostics anuyk20v-diagnostics cv3333u-diagnostics on143vusq-diagnostics rd397u-diagnostics		1

Table B-3 (Continued)

Data Name		Description		Notes
menu-id	=	tt624v5ug-diagnostics ref-des-test-menu rd-test-types replace-lru-menu equipment-lru-menu anush26v-episodes anusq69v-episodes anuyk20v-episodes cv333u-episodes on143vusq-episodes rd397u-episodes tt624v5ug-episodes field-operating-tests control-circuits-tests step-procedures pin-number-test-menu pin-number-test-types student-menu}		
menu-node	=	menu-id 0 {option-node} n	+ 	7
menu-tree	=	*0 {menu-node} n*		8
option-node	=	branch-code num-menu-stack-pops branch-function-name	+ + 	2 3 4
rd-test-status	=	equipment-type rd rd-test-type return-menu	+ + + 	9
uyk20-status	=	register-access		10
step-status	=	step-category step-number return-menu	+ + 	11
fot-status	=	test-name		12

1. Each menu-id corresponds to one PANEL Plus file, which holds one screen menu. The panel filenames for each menu-id are as follows:

menu-id	panel filename holding menu
administrator-menu	adminmen.pnl
student-menu	stdntmen.pnl
equipment-menu	equipmen.pnl
	eqmenurj.pnl (right-justified)
get-episode-menu	getepmen.pnl
system-level-menu	syslevme.pnl
fallback-tests-menu	fallbmen.pnl
load-operational-menu	lopmenu.pnl
replace-menu	replacem.pnl
anush26v-episodes	ush26eps.pnl
anusq69v-episodes	usq69eps.pnl
anuyk20v-episodes	uyk20eps.pnl
cv3333u-episodes	cv333eps.pnl
on143vusq-episodes	on143eps.pnl
rd397u-episodes	rd397eps.pnl
tt624v5ug-episodes	tt624eps.pnl
anush26v-tests	ush26tst.pnl
anusq69v-tests	usq69tst.pnl
anuyk20v-tests	uyk20tst.pnl
cv3333u-tests	cv333tst.pnl
on143vusq-tests	on143tst.pnl
rd397u-tests	rd397tst.pnl
tt624v5ug-tests	tt624tst.pnl
anush26v-diagnostics	h26diags.pnl
anuyk20v-diagnostics	k20diags.pnl
cv3333u-diagnostics	cv3diags.pnl
on143vusq-diagnostics	on1diags.pnl
rd397u-diagnostics	rd3diags.pnl
tt624v5ug-diagnostics	t24diags.pnl
ref-des-test-menu	rdmenu.pnl
rd-test-types	rdtstmen.pnl
pin-number-test-menu	pinmenu.pnl
pin-number-test-types	pintstme.pnl
step-procedures-menu	stepmenu.pnl
field-operation-test-menu	fottests.pnl
control-circuits-menu	ctrlcirc.pnl

2. Branch-code is interpreted in the following way:

Branch-code	Meaning
0-n	Branch to the menu in the menu-tree that is indexed by branch-code.
-1	Terminate the menu driver process. When selected, the program loop presenting menus is exited.
-2	Call the function with name branch-function-name.

3. Num-menu-stack-pops indicates the number of times the menu clear function should be called if the option is selected. This has the effect of "popping" off overlaid menus.

4. Branch-function-name is the name of the function to call when branch-code is -2.

5. Each form-id corresponds to one PANEL Plus file, which holds one screen form. The panel filenames for each form-id are as follows:

form-id	panel filename holding menu
instructor-exit-form	exitform.pnl
a124-step-number-form	a124form.pnl
password-form	password.pnl
rd-test-form	rdform.pnl
pin-number-test-form	pinform.pnl
replace-rd-form	reprdfrm.pnl
ssn-form	ssnform.pnl
uyk20-step-form	stepform.pnl
ush26-step-form	h26stfrm.pnl
usq69-step-form	q69stfrm.pnl

6. This data item holds presentation status for current session.

7. Each menu node contains the panel file name holding the menu (menu-id), plus an array of option-nodes describing the options from the menu.

8. The menu-tree defines the menu hierarchy of the PRESENT program.

9. Rd-test-status temporarily holds reference designator and pin number information when rd tests or replacements are being performed.

10. Uyk20-status tells whether the troubleshooter has gained register access.

11. Step-status temporarily holds USH26 and USQ 69 step procedure status information.

12. Fot-status temporarily holds USH26 field operating test status information.

Package Dependencies

Figure B-4 illustrates the software packages that PRESENT depends on to do its job. PRESENT uses the TAE panel_pl code package to display menus and forms as well as miscellaneous screen management functions. PRESENT uses the TAE graphics code package to display Dr. Halo graphics and the TAE tpep_dbs code package to access the episode and student history databases.

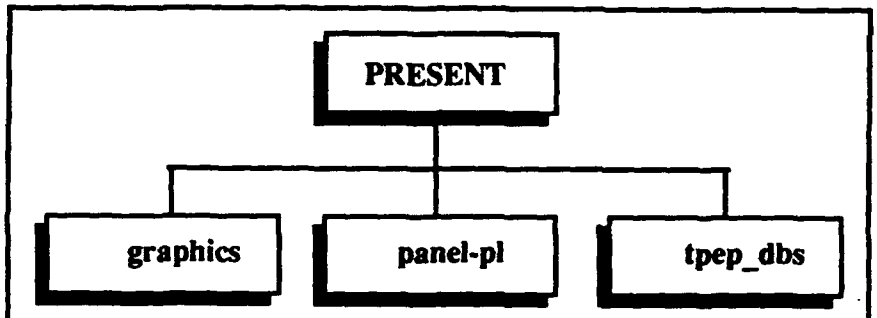


Figure B-4. PRESENT Package Dependencies.

EDITVIEW PROGRAM

Description

The EDITVIEW program has a menu-driven user interface. EDITVIEW is implemented in code as eight separate compilation packages. One of these packages, Edithist.pak, uses a menu-tree structure as in PRESENT for presenting the menus to the user. The other packages do not use a menu-tree structure, but call up the various menus within the individual routines. Here are the names and functions of the eight compilation packages:

1. Editview.pak contains the main routines for invoking the various features available to the user and for driving the "print multiple histories" feature. This package also has the routines for deleting a student from the student history database and for implementing the display summarizing each student's final scores, and for displaying the overall class average score.

2. Scord_s.pak contains the code for calculating, displaying, and printing a student's scored analysis for a given episode history.

3. Author_s.pak contains the code for allowing the user to author scoring constants and weights to be used in conducting the scored analysis of a sequence of events.

4. Sequence.pak contains the code for displaying and printing one sequence of events.

5. Archival.pak contains the code for archiving student history databases and retrieving them.

6. Combine.pak contains the code for transferring a student history database to a floppy diskette and for combining two different student history databases into one database.

7. Edithist.pak contains the routines for allowing Navy instructor to edit a student's sequence of events.

8. Statistic.pak contains the routines for translating the data in the student history database into a data file for the Microstat statistical analysis software package.

EDITVIEW Abstract Data Dictionary

The data dictionary shown in Table B-4 defines data global to the NAVMACS EDITVIEW program. The numbered note listed below refers to the number in the notes column in Table B-4.

Table B-4

EDITVIEW Abstract Data Dictionary

Data Name	Description	Notes
MAXSTUDENTS	= 75	
ssn-list	= 0{ssn}MAXSTUDENTS	1

1. SSN list is a list of all the unique social security numbers in the current student history database.

Package Dependencies

EDITVIEW uses the TAE panel_pl code package to display various menus and forms for user interaction. It uses the TAE tpep_dbs code package to access and change the student history database. An exception is with the routines which combine (merge) two student history databases. They do not use the TAE tpep_dbs code package. Instead, they call c-tree library routines directly.

TAE panel_pl CODE PACKAGE

Description

The panel_pl code package provides an interface to the PANEL Plus screen management tools. The primary screen management functions are (1) menu display and interaction, and (2) form display and interaction. Menu display and interaction is handled by the function "menu_interaction"; form display, by the function "form_draw"; and form interaction, by the function "form_interaction". Figure B-5 illustrates a high-level data flow diagram of the TAE panel_pl code package.

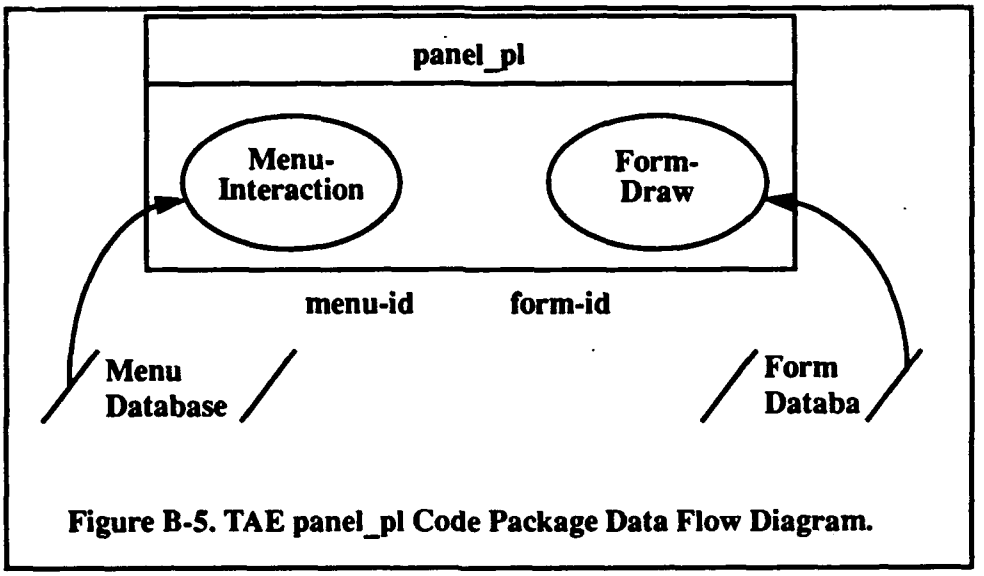


Figure B-5. TAE panel_pl Code Package Data Flow Diagram.

The TAE panel_pl code package maintains a "menu-stack" of currently displayed menus and allows a maximum of three menus to be on the screen at any one time. Menus are pushed onto the stack with a call to "menu_interaction". Menus are popped off of the stack (and the screen) with a call to "menu_pop".

Only one form may be displayed at a time. However, forms may be overlaid on screens that contain menus.

TAE panel_pl Code Package Abstract Data Dictionary

The data dictionary shown in Table B-5 defines data global to the TAE panel_pl code package. The numbered notes listed below refer to the numbers in the notes column shown in Table B-5.

Table B-5

TAE panel_pl Code Package Abstract Data Dictionary

Data Name		Description		Notes
form	=	panel-plus-control-block		
menu-definition	=	panel-plus-control-block	+	1
		internal-field-control-block	+	
		current-selection		
menu-stack	=	top	+	2
		0 {menu-definition} NUM-OVERLAYS		

1. Each menu-definition is represented in two formats. The higher-level format (panel-plus-control-block) describes the characteristics of the panel file (".pnl" file) that holds the menu. The lower-level format (internal-field-control-block) describes the menu itself in specific detail. By convention, each PANEL Plus file that will hold a menu has only one field: a multiline field that will act as a menu. panel-plus-control-block and internal-field-control-block are panel plus data structures. Their coded structure names are "panpcb" and "ifcb", respectively. They are described in greater detail in the PANEL Plus user's manual. Current-selection is used to hold the latest selection made from the menu.

2. The menu-stack holds all currently displayed menus. If only one menu is currently displayed, the menu-stack will only have one entry. The menu-stack grows as menus are overlaid on top of one another. There is a limit to the size of the menu-stack, defined by NUM-OVERLAYS. For correct stack management, menu pushes must be matched with menu pops. The top of the menu-stack will point to the next open slot of the menu-stack. The stack is empty when top of menu-stack equals zero. The stack is full when top of menu-stack = NUM-OVERLAYS.

PANEL Plus Conventions

The TAE panel_pl code package is implemented with the PANEL Plus commercial software package. The PANEL Plus package is a character-based form and menu package. On the target hardware, it operates with the EGA card in text mode 3, providing a character resolution of 80 x 25, with 16 foreground and 8 background colors. The following lists specific conventions adopted with PANEL Plus:

1. TAE uses the pre-configured version of PANEL Plus for the IBM PC. Since this pre-configuration is used, the "tailor" program is not necessary. The name of the screen editor for the IBM version of PANEL Plus is "panelpi". The name of the screen test program for the IBM version of PANEL Plus is "pantesti".

2. Colors are defined by a DOS environment variable, PAIATT, set with the following command:

   "set PAIATT=1F711C797B7F060708090A0B0C0D0E0F".

3. This string is interpreted as 16 2-digit, unsigned hexadecimal numbers, corresponding to the 16 highlight types (colors) provided by the PANEL Plus editor. The highlight types defined for TAE, as generated by the values of the PAIATT variable are listed below. The *PANEL Plus User's Manual describes the correspondence of hexadecimal numbers to colors* for the IBM PC.

Highlight Type	PAIATT Value	Color
0	1F	Bright White On Blue
1	71	Blue On White
2	1C	Bright Red On Blue
3	79	Bright Blue On Grey
4	7B	Bright Cyan On Grey
5	7F	Bright White On Grey
6	06	Brown On Black
7	07	Light Grey (White) On Black
8	08	Dark Grey On Black
9	09	Light Blue On Black
10	0A	Light Green On Black
11	0B	Light Cyan On Black
12	0C	Light Red On Black
13	0D	Light Magenta On Black
14	0E	Yellow On Black
15	0F	Bright White On Black

4. The first six highlight types have the following special meaning:

   - Highlight type 0 (bright white on blue) = the default screen color attributes.
   - Highlight type 1 (blue on white) = inverse video from the default screen color attributes.
   - Highlight type 2 (light red on blue) = color of the menu highlight bar.
   - Highlight type 3 (bright blue on grey) = color of an event "tagged" for deletion in the edit mode of EDITVIEW.
   - Highlight type 4 (bright cyan on grey) = color of a "highlighted" event which is also "tagged" for deletion in the edit mode of EDITVIEW.
   - Highlight type 5 (bright white on grey) = color of a "highlighted" event during deletion of events in the edit mode of EDITVIEW.

In addition to these 16 highlight types, other color effects can be generated using the panel editor's "highlight modifiers". The highlight modifiers defined for TAE are listed below. Highlight modifiers can be added together, creating still more color effects. For example, a highlight modifier of 224 (128 + 96) would blink the foreground (128) on top of a yellow background (96).

Highlight Modifier	Effect
0	None
16	Blue Background
32	Green Background
48	Cyan Background
64	Red Background
82	Magenta Background
96	Yellow Background
112	White Background
128	Blinking Foreground

Menus are built by creating a multiline field with the panel editor. Each line of the field corresponds to an option from the menu. Each menu must reside in its own panel file. The menu must be authored on level zero of the panel.

Forms are also created with the panel editor, one form per file. The fields of the form must be authored on level one of the panel. A border surrounding the entire form must be authored on level two of the panel.

TAE displays menus and forms using the "dynamic load" function (paload) of PANEL Plus.

Application programs must link to the special IBM library, "panelpi.lib", as well as the usual PANEL Plus library, "panelp.lib". In the link statement, "panelpi.lib" should be placed before "panelp.lib".

TAE GRAPHICS CODE PACKAGE

The purpose of this package is to display Dr. Halo, 640 x 350, 16-color graphic images. To display the image, the package must be able to switch the EGA card from text mode 3 (80 x 25; 16 foreground colors; 8 background colors) to graphics mode 16 (640 x 350; 16 colors). In addition to displaying a graphic, this package must be able to save the text image that was on the screen before the graphic display request and then to restore that text image after the graphic display.

Saving the currently displayed text image is handled by function "ega_savetext". Displaying a Dr. Halo picture file is handled by function "ega_graphic". Restoring the saved text image is handled by function "ega_restoretext". Figure B-6 illustrates a high-level data flow diagram of the TAE graphics code package.

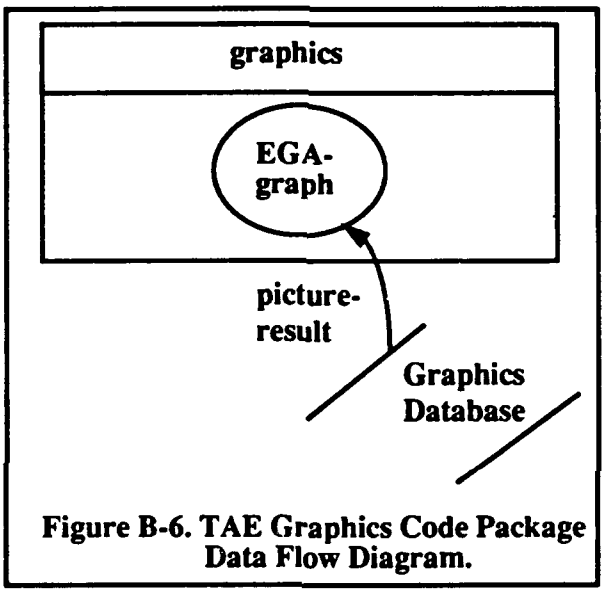


Figure B-6. TAE Graphics Code Package Data Flow Diagram.

This package also provides an interface to the MetaWINDOW/PLUS graphic drawing routines. Presently, MetaWINDOW/PLUS is used only to draw simple graphics over Dr. Halo images (for example, a box with some status information). The graphics package does not maintain any major data structures.

The graphic drawing routines of the graphics package are implemented with the MetaWINDOW/PLUS commercial graphics package. The following lists implementation conventions adopted with MetaWINDOW/PLUS:

1. MetaWINDOW/PLUS is set to EGA, 640 x 350, with 16 colors.

2. The graphics origin, pixel coordinate (0, 0), is set to the upper-left corner of the screen (some applications will set it to the lower-left corner).

3. The default graphics pen size has been set to be 3 pixels by 3 pixels.

4. The default EGA font, "system16.fnt", is used for displaying text.

TAE DATABASES AND THE TAE tpep_dbs CODE PACKAGE

TAE Databases

The TAE databases are divided into an episodes database and a student history database, as illustrated in Figure B-7. The episode database is a depository for the troubleshooting episodes created by the ADDEP program and delivered by the PRESENT program. The student history database is a depository for the student histories generated by the PRESENT program and viewed by the EDITVIEW program

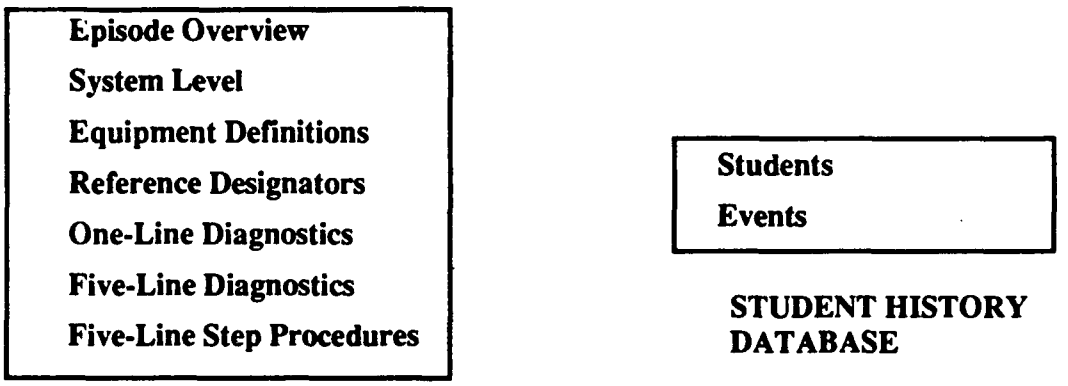


Figure B-7. TAE Databases.

Physically, the episode database is divided into seven databases: episode overview, system level, equipment definitions, reference designators, one-line diagnostics, five-line diagnostics, and five-line step procedures. The physical structure of the episode database was designed to reflect the "levels of analysis" of the NAVMACS system: system level, equipment level, and component level. These "levels of analysis" are described in the TAE SRD. Figure B-8 illustrates the mapping between the logical view of the NAVMACS system and the physical structure of the episodes database.

The administration level is a new level containing episode management information. The episode overview database stores administration level information. NAVMACS system-level information is stored in the system-level database. NAVMACS equipment-level information is stored in the equipment definitions database. NAVMACS component-level information is stored in the reference designators, one-line diagnostics, five-line diagnostics, and five-line step procedures databases.

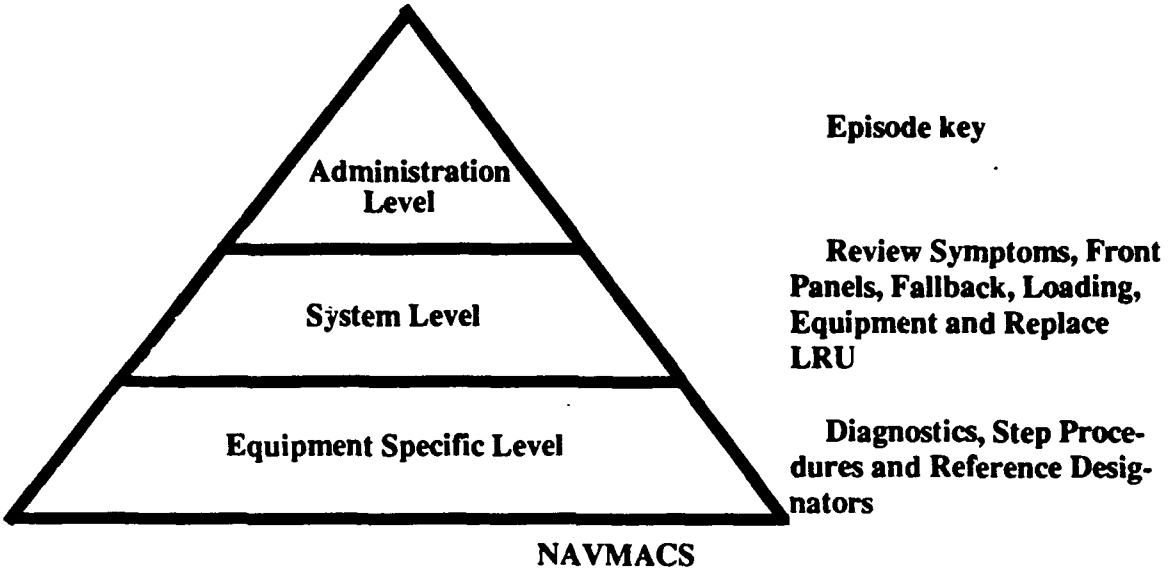


Figure B-8. Logical to Physical Mapping.

Each of the episode databases is built from fixed-length data records. For a particular episode, the data records defining the episode are spread out among the episode databases. There is one episode overview data record per episode. There are nine system-level data records per episode (six fallback tests and three load operational program tests). There are eight equipment definition data records per episode (one for each piece of equipment). The number of component-level data records is a variable, and depends on the particular episode. Figure B-9 provides an example the data records required for an AN/UYK-20 (V) and a TT-624 (V) 5/UG episode.

Physically, the student history database is divided into two parts: students and events. The student's portion contains a record for each student in the current class who has operated the PRESENT program. These data records contain the student's social security number and episodes completed record. The events portion holds all events generated by the PRESENT program. The EDITVIEW software builds logically sound student histories from the events database.

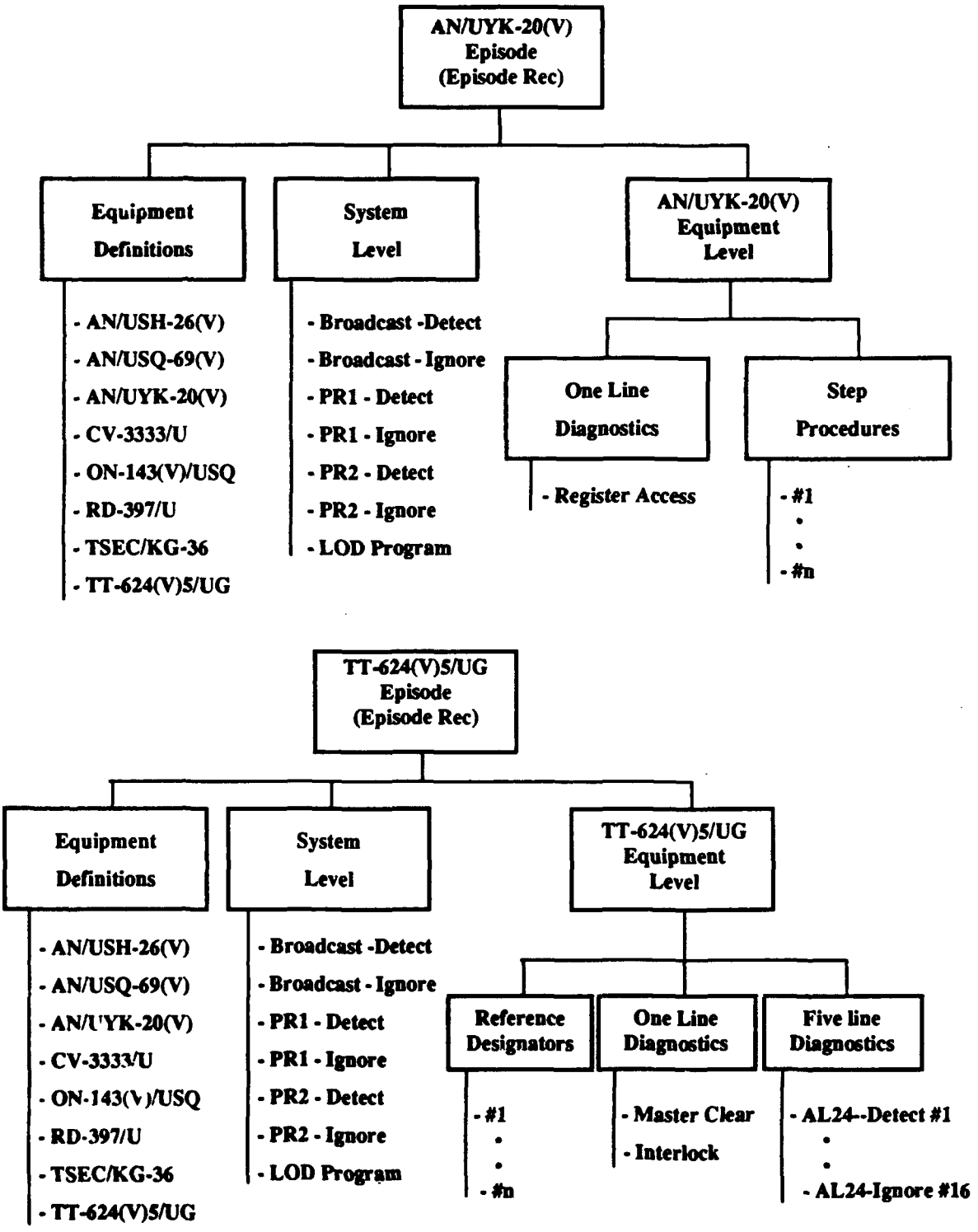


Figure B-9. AN/UYK-20 (V) and TT-624 (V) 5/UG Episodes.

Figure B-10 illustrates three levels of layering used to access the TAE databases. The TAE tpep_dbs code package provides ADDEP, PRESENT, and EDITVIEW with access to the TAE databases. At the lowest layer, the c-tree software package coordinates the interaction with the databases. c-tree uses the parameter file "tpep.prm" as a definition of the structure of the databases. During the combining of student history databases, a feature under EDITVIEW, c-tree uses the parameter file "combine.prm", which contains the structure of the student history databases being combined.

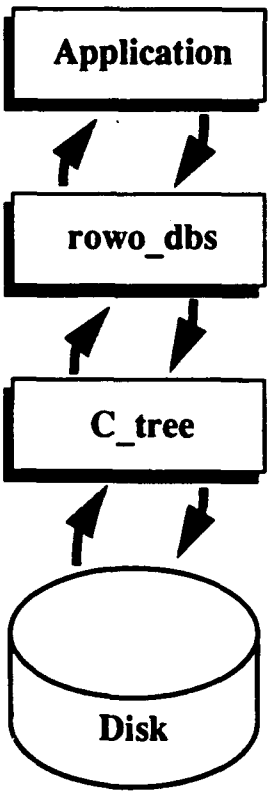


Figure B-10. Database Layering.

The episode database is stored in seven c-tree data files (one for each type of data). Each record in each data file is of a fixed length. Each data file is accessed by an episode-key, as well as possibly by other keys. depending on the data file. The episode-key is defined in the NAVMACS TAE Abstract Data Dictionary shown in Table B-1. Table B-6 lists each data file in the episodes database, the data file's access keys, and a reference into the data dictionary to the data item describing the data file's record structure.

Table B-6

Episode Database Division

Data Filename	Access Keys	Data Dictionary Data Item
episode.dat	episode-key	episode-data-rec
equip.dat	episode-key equipment-type	equip-data-rec
syslev.dat	episode-key test-name	system-level-data-rec
refdes.dat	episode-key rd rd-test-type	rd-data-rec
diag.dat	episode-key diagnostic-name diagnostic-number	diag-data-rec
fvlndiag.dat	episode-key diagnostic-name diagnostic-number	five-line-diag-rec
fvlnstep.dat	episode-key step-category	five-line-step-rec

The student history database is divided into two c-tree data files (one for each type of data). Each record in each data file is of a fixed length. Table B-7 lists each data file in the student history database, the data file's access keys, and a reference into the data dictionary to the data item describing the data file's record structure.

Table B-7

Student History Database Division

Data Filename	Access Keys	Data Dictionary Data Item
events.dat	student-key episode-key sequence-num	event-record
students.dat	ssn	student-rec

The episode and student history databases are opened by a call to the function "open_databases". The "combine" feature under EDITVIEW opens the student history databases with a call to function "open_dbs". Each data file is associated with two functions: "read-data-record" and "write-data-record". "Read-data-record" reads one data record from the specified data file; "write-data-record" writes one data record to the specified data file. The episode and student history databases are closed by a call to function "close_databases". The "combine" feature closes its student history database files with a call to "close_dbs". Figure B-11 illustrates a high-level data flow diagram of the TAE tpep_dbs code package.

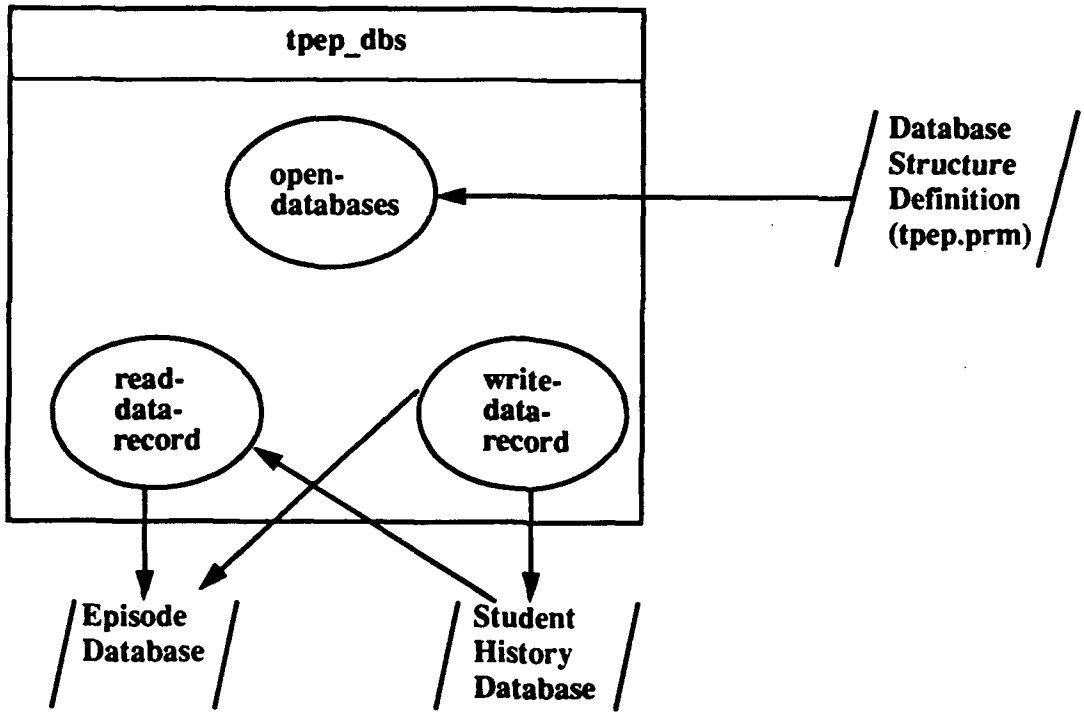


Figure B-11. TAE tpep_dbs Code Package Data Flow Diagram.

The TAE tepe_dbs code package also exports a function called "vf_close", standing for virtual file close. One quirk with c-tree is that it hogs DOS file descriptors. Calling "vf_close" forces c-tree to free up a file descriptor, so that an application program may perform disk input/output independent of c-tree.

TAE tpep_dbs Code Package Abstract Data Dictionary

The data dictionary shown in Table B-8 defines data global to the TAE tpep_dbs code package. The numbered notes listed below refer to the numbers in the notes column shown in Table B-8. Refer to the NAVMACS TAE Abstract Data Dictionary in Table B-1 for many of the data types not defined here.

Table B-8

TAE tpep_dbs Code Package Abstract Data Dictionary

Data Name		Description		Notes
diag-event-rec	=	diagnostic-test	+	
		evaluation	+	
		test-number	+	
		proof-pt-status		
episode-data rec	=	episode-key	+	
		faulted-rd	+	
		symptoms	+	
		good-fault-replacements	+	
		presentation-mode	+	
		min-num-tests		
equip-data-rec	=	episode-key	+	
		equipment-type	+	
		evaluation	+	
		front-panel-graphic	+	
		maintenance-panel-graphic		
equip-select-event-rec	=	equipment-type	+	
		evaluation		
event-record	=	student-key	+	
		episode-key	+	
		sequence-num	+	
		time	+	6
		event-type	+	
		[login-event-rec	|	
		logout-event-rec	|	
		equip-select-event-rec	|	
		fallback-event-rec	|	
		front-panel-event-rec	|	
		load-program-event-rec	|	
		maint-panel-event-rec	|	
		rd-test-event-rec	|	
		replace-event-rec	|	
		review-event-rec	|	
		iexit-event-rec	|	
		diag-event-rec	|	
		step-event-rec	|	
		revision-event-rec]	|	

Table B-8 (Continued)

Data Name		Description		Notes
fallback-event-rec	=	fallback-tests	+	
		parity-switch	+	
		evaluation		
five-line-diag-rec	=	episode-key	+	
		diagnostic-test	+	
		test-number	+	
		evaluation	+	
		proof-pt-status	+	
		five-line-result		
five-line-step-rec	=	episode-key	+	
		step-category	+	
		step-number	+	
		evaluation	+	
		proof-pt-status	+	
		five-line-result		
front-panel-event-rec	=	equipment-type	+	
		evaluation		
iexit-event-rec	=	exit-reason		3
load-program-event-rec	=	load-operational-program-tests	+	
		evaluation		
login-event-rec	=			4
logout-event-rec	=			4
maint-panel-event-rec	=			4
one-line-diag-rec	=	episode-key	+	
		diagnostic-test	+	
		test-number	+	
		evaluation	+	
		proof-pt-status	+	
		text-result		
rd-data-rec	=	episode-key	+	
		rd	+	
		proof-pt-status	+	
		1 {rd-test-result}	NUM-TESTS-PER-RD	

Table B-8 (Continued)

Data Name		Description		Notes
rd-event-rec	=	rd	+	
		rd-test-type	+	
		evaluation	+	
		proof-pt-status		
rd-test-result	=	rd-test-type	+	
		evaluation	+	
		text-result		
replace-event-rec	=	equipment-type	+	
		rd	+	
		evaluation		
review-event-rec	=			4
revision-event-rec	=	revision-number	+	
		reason		
step-event-rec	=	step-category	+	
		step-number	+	
		evaluation	+	
		proof-pt-status		
student-key	=	integer		5
student-rec	=	ssn	+	
		episodes-completed-rec	+	2
		student-key		
system-level-data-rec	=	episode-key	+	
		[fallback-tests	|	
		load-operational-program-tests]	+	
		evaluation	+	
		text-result		

2. The episodes completed record is an array of bytes that keeps a record of which episodes a student has completed. There is one episodes completed record per student. NUM_EPISODES holds the total number of episodes available in the TAE system.

3. An instructor exit event is recorded when the instructor has prematurely terminated a presentation session. The instructor exit event holds the reason the instructor terminated the session.

4. This event type requires no additional status information; it is sufficient just to list that the user performed the event.

5. Student-key is a unique id number for a student. It will be the last four digits from the student's social security number.

6. Event time resolution is only kept to minutes. Both hours and minutes will be positive integers. The first event occurs at time zero hours, zero minutes. Minutes range from zero to 59.

c-tree Conventions

The TAE tpep_dbs code package is implemented using routines provided in the c-tree commercial software package. The c-tree software package is compiled into two different library files:

1. ctreefst.lib, which buffers index file updates and writes them to disk as needed.

2. ctree.lib, which immediately forces all index file updates to disk. Applications linked with ctree.lib will run slower than applications that are linked with ctreefst.lib.

This is a list of implementation conventions adopted with c-tree:

1. Only the ISAM (Indexed Sequential Access Method) routines are used.

2. All data records are of fixed length (no variable-length data records).

3. The c-tree parameter file "tpep.prm" describes the structure of all TAE databases.

4. The c-tree parameter file "combine.prm" describes the structure of the student history databases that it merges together.

5. The PRESENT and ADDEP programs are linked with ctree.lib.

6. The EDITVIEW program is linked with ctreefst.lib to speed up execution.

TAE LOCAL AREA NETWORK DESIGN AND IMPLEMENTATION

The Omninet local area network (LAN) by Corvus Systems is used to create a networking environment for TAE. each computer connected to the LAN is referred to as a "node". Corvus Systems provides its PC/NOS network operating software as part of Omninet. PC/NOS must be installed on each computer before the computer is attached to the LAN. The Omninet hardware must also be installed in each computer, and cables must be connected to the computers to create the connections that form the physical portion of the LAN. Once the Omninet software and hardware are properly installed, logical connections between the nodes must be defined to allow for hard disk sharing. The Omninet LAN together with custom TAE batch files affords the following advantages over using TAE without them:

1. An instructor working at one node may specify troubleshooting episodes for specific students to run at the remote nodes.

2. An instructor working at one node may access and edit student history databases located at remote nodes.

The following steps explain how to set up the Omninet LAN:

1. Connect each computer to the LAN using the Corvus hardware. This involves (a) installing the Omninet Transporter Network Interface Card according to the instructions in the Corvus Installation and User Guides booklet and (b) installing the cabling according to the instructions in the Omninet Cabling System II Installation Guide.

2. Install the PC/NOS network operating system software on ALL nodes in the "shared disk" configuration according to the instructions in the "Installation" section of the PC/NOS System Manager's Guide.

3. Designate one node as the "administrator node". The classroom instructor will use this computer to deliver troubleshooting episodes to the students on their remote nodes, and may also access the student history databases on remote nodes for use with the EDITVIEW program.

4. Create connections to all remote hard disks on the LAN from the administrator node. Each computer on the network should refer to its own hard disk as "C:". The hard disk drives on remote machines are referred to as drives "D:" through "M:". The hard drive known as "D:" to a remote machine will be referred to as "D:" by all remote machines; likewise for drives "E:" through "M:". If a computer already refers to a partition on its hard disk as the "D:" drive, make sure not to use "D:" as one of the network drive names for referring to remote machines. These logical connections (known as "plugs and sockets" under PC/NOS) can be made using the PC/NOS Netview program. Refer to the "Making Connections" section in the PC/NOS System Manager's Guide to create these plugs and sockets. If the administrator node ever experiences a crash, the instructor may use any other node as an administrator node, provided that the substitute node also has plugs to every other hard disk on the LAN. For maximum flexibility, configure every node so that it has plugs to every other node's hard disk, following the naming scheme explained above.

5. Invoke the PC/NOS operating system once the LAN is installed. Refer to the "General Tasks" section of the PC/NOS System Manager's Guide to log on and log off the LAN at each node.

6. Move all four files from the C:\TPEP\RUNTIME\OMNINET subdirectory to the C:\TPEP\RUNTIME subdirectory. There are three batch files and one executable file.

The following steps explain how to use the TAE programs under the Omninet LAN:

1. Have the instructor set up troubleshooting episodes for a remote machine using two batch files under the C:\TPEP\RUNTIME subdirectory, NETFILE.BAT and SEND.BAT. Run NETFILE to specify the student's Social Security Number (SSN) and to specify the episodes the student is to perform. This batch file invokes the mknetfil.exe executable program. Run SEND to send that information to the remote node the student is using. To send the information to drive E: type "send E:" and press ENTER.

2. Have the student run the PRESENT program. The student types "present" and presses ENTER to start the program. The PRESENT program will recognize whether the instructor has sent a list of troubleshooting episodes for the student to perform. If that list exists on the student's node, PRESENT will automatically run the episodes for the student. If no list was sent, PRESENT will prompt the operator to enter the student's SSN and troubleshooting episodes with the usual menu-driven interface.

3. Have the instructor run EDITVIEW on the administrator node using any of the student history databases located on the remote nodes. The EV_NET.BAT batch file exists for this purpose. Then type "ev_net" followed by the name of the remote drive and press ENTER. The student history database on the local drive will be saved, the database on the remote drive will be copied to the administrator node's drive, and the EDITVIEW program will run. The student history database on the remote machine will be locked while the instructor runs EDITVIEW; that is, the student will not be able to run PRESENT. When the instructor quits EDITVIEW, the database will be copied back to the drive it came from and the original database on the local drive will be restored. If no drive name is specified, the local student history database will be used.

APPENDIX C

TROUBLESHOOTING ASSESSMENT AND ENHANCEMENT (TAE) SOFTWARE DISKETTES: USE

Note. Volume 2 contains the diskettes.

TAE SOFTWARE DISKETTES

Since the project name used to be "TPEP", several directory names and file names associated with this project use the acronym "TPEP". To install the system on a hard disk, the same directory structure that is on the delivery diskettes must be used.

Two groups of diskettes (in Volume 2) accompany this appendix: source code diskettes and runtime environment diskettes. The runtime environment diskettes are labeled "C:\TPEP\RUNTIME runtime environment" and "C:\TPEP\ADDEP runtime environment". The source code diskettes are labeled "TAE SOURCE CODE".

This appendix explains what is on these diskettes and how to use them. All diskettes delivered are in standard MS-DOS 360- kilobyte format ("double density").

All software was developed using the following Microsoft (R) products designed to run under MS-DOS (R):

1. Microsoft (R) C Compiler Version 4.00
   Copyright (C) Microsoft Corporation 1984, 1985, 1986

2. Microsoft (R) Macro Assembler Version 4.00
   Copyright (C) Microsoft Corporation 1981, 1983, 1984, 1985

3. Microsoft (R) Overlay Linker Version 3.51
   Copyright (C) Microsoft Corporation 1983, 1984, 1985, 1986

4. Microsoft (R) Library Manager Version 3.04
   Copyright (C) Microsoft Corporation 1983, 1984, 1985, 1986

Installing the Runtime Environment Diskettes

There are two runtime environments, manifested in two separate directories:

1. \TPEP\RUNTIME\, which has the PRESENT, EDITVIEW, and NETFILE programs and all necessary files to support them.

2. \TPEP\ADDEP\, which has the ADDEP, DELEP, and CREATEDB programs and all files to support them.

To install the runtime environments, perform the following steps:

1. On your Z-248 computer, create a directory called "TPEP" on your C: drive. The TAE system assumes that you have a hard disk named "C:", which you must have for all features to run correctly.

2. Create a subdirectory under C:\TPEP called "RUNTIME".

3. To transfer files from the diskettes into the C:\TPEP\RUNTIME subdirectory, make sure you are logged into that subdirectory, then place the first diskette for \TPEP\RUNTIME in the A: floppy drive. Type "copy A:*.* " and press ENTER. Continue copying files from the remaining TPEP\RUNTIME diskettes. The PRESENT and EDITVIEW programs can now be run from this directory.

4. Create a subdirectory under C:\TPEP called "ADDEP".

5. To transfer files from the diskettes into the C:\TPEP\ADDEP subdirectory, make sure you are logged into that subdirectory, then place the first diskette for \TPEP\ADDEP in the A: floppy drive. Type "copy A:*.* " and press ENTER. Continue copying files from the remaining TPEP\ADDEP diskettes. The ADDEP, DELEP, and CREATEDB programs can now be run from this directory.

For a complete listing of the files that should be in each subdirectory, see the README file in each subdirectory.

Using the Source Code Diskettes

The source code is organized on the delivery diskettes in a directory structure that matches the development environment. The following is a template of the directory structure used: <project-name>\<program name>\<package name>\<file names>. Two examples of the directory structure are provided:

TPEP.PRJ\PRESENT.PRG\PRESENT.PAK\source & object files
TPEP.PRJ\TPEP.LIB\INTRUPT.PAK\source & object files

The TAE project is made up of five programs and the TPEP library. Each of these is made up of one or more code packages. All the source code for one package is held in one subdirectory. The name of that subdirectory is the name of the code package. There are also "interface" subdirectories named "INTRFC" that hold C language header files for inclusion into one or more code packages. In general, each code package is a separate compilation unit and has only one file that gets compiled. This file has the extension ".c" and "#includes" all the other source files for that code package. These other source files all have the extension ".cf".

Once a code package is compiled, it needs to be linked with other object and library (.lib) files using the Microsoft Overlay Linker to produce an executable file (a runnable program). The TPEP library code packages are run through the Microsoft Library Manager instead of the Microsoft Overlay Linker. This produces the TPEP library that the other executable files must link to.

Source code for the CREATEDB program is not provided. This is a simple utility program used to create a new empty set of student history and episode database files. This program resides under the TPEP\ADDEP\ subdirectory and is not normally used. If you run this program you must then run ADDEP for each episode that you want to have in the episode database.

Producing Executables

Here is how the programs and packages are organized and how to produce the executables and library files:

1. ADDEP

The ADDEP program has one code package that contains all source code for the ADDEP executable program. Create one object (.obj) file by compiling the addep.c file, which "#includes" all <file name>.cf files. Link the package with the other necessary object and library files by invoking the Microsoft Overlay Linker with the command "link @addep.lnk".

2. DELEP

The DELEP program has one code package that contains all source code for the DELEP executable program. Create one object file by compiling the delep.c file, which "#includes" all <file name>.cf files. Link the package with the other necessary object and library files by invoking the Microsoft Overlay Linker with the command "link @delep.lnk". The program is executed in the runtime environment by calling the DELEP.BAT batch file.

3. PRESENT

The PRESENT program has one code package that contains all source code for the PRESENT executable program. Create one object file by compiling the present.c file, which "#includes" all <file name>.cf files. Link the package with the other necessary object and library files by invoking the Microsoft Overlay Linker with the command "link @present.lnk". Once present .exe is created, rename it to presnt.exe so that it will be invoked in the runtime environment by calling the PRESENT.BAT batch file.

4. EDITVIEW

There are eight code packages under EDITVIEW and one INTRFC package. The code package names and contents are described in Appendix B. Create one object file for each package by compiling the <package name>.c file, which "#includes" all <file name>.cf files in that package. Link together all eight object files along with the other necessary object and library files by invoking the Microsoft Overlay Linker with the command "link @editview.lnk" under the EDITVIEW.PAK package. Once editview.exe is created, rename it to edview.exe so that it will be invoked in the runtime environment by calling the EDITVIEW.BAT batch file.

5. MKNETFIL

The MKNETFIL program has one code package that contains all source code for the MKNETFIL executable program. Create one object file by compiling the mknetfil.c file, which "#includes" all <file name>.cf files. Link the package with the other necessary object and library files by invoking the Microsoft Overlay Linker with the command "link @mknetfil.lnk". This program is executed in the runtime environment by calling the NETFILE.BAT batch file.

6. TPEP library

The TPEP library is a collection of routines that may be used by all of the TAE executable programs. Six code packages and one interface package make up TPEP library:

a. GRAPHICS.PAK--graphics-related functions.

b. INTERUPT.PAK-- functions for substituting the MS-DOS keyboard interrupt handler.

c. PANEL_PL.PAK--functions that interface to the PANEL Plus form and menu-handler commercial software package.

d. PRINTUTI.PAK--functions for interfacing to a printer.

e. TOOLS.PAK--miscellaneous functions commonly needed by all TAE programs.

f. TPEP_DBS.PAK-- functions that interface to the c-tree database-handler commercial software package.

g. INTRFC--not a true code package: contains only header files common to the other packages.

Compile each TPEP library code package separately. Except for the INTERUPT.PAK package, each code package contains several C language source files. Create one object file for each package by compiling the <package name>.c file, which "#includes" all <file name>.cf files. The INTERUPT.PAK package contains two C language source files, each of which must be compiled separately, and three assembly language files that must be assembled separately using the Microsoft Macro Assembler. The PRINTUTI.PAK package also contains one assembly language file (lptactiv.asm) that must be assembled separately. Create the TPEP library by running all resulting object files through the Microsoft Library Manager.

Creating the Runtime Environments

Create the executables as outlined above. Then copy them to the respective runtime subdirectories. All other support files to these executables are data files, batch files, or database files that cannot be re-created from the source code files.

Library and Object Files Needed From Commercial Software Packages

Each TAE executable program needs to be linked with several of the following object and library files that were generated directly from the commercial software packages purchased for and used by the TAE project:

1. pafr.obj

This is the object file created by compiling a modified version of pafr.c. The original pafr.c was taken from the PANEL Plus package. The source code and object file are both found under TPEP.PRJ\PANEL\on the source code diskettes.

2. pamsge.obj

This is the object file created by compiling a modified version of pamsge.c. The original pamsge.c was taken from the PANEL Plus package. The source code and object file are both found under TPEP.PRJ\PANEL\on the source code diskettes.

3. panelpi.lib

This is created by following the instructions provided with the PANEL Plus package. It is found under TPEP.PRJ\PANEL\ on the source code diskettes.

4. panelp.lib

This is created by following the instructions provided with the PANEL Plus package. It is found under TPEP.PRJ\PANEL\ on the source code diskettes.

5. ctreefst.lib

This is created by building the c-tree package for use under an MS-DOS system according to the instructions provided with the c-tree package. It is found under TPEP.PRJ\CTREE\ on the source code diskettes.

6. ctree.lib

This is created by building the c-tree package for use under an MS-DOS system using the FPU-TONLY and DOSFLUSH options defined in the CTOPTN.H file according to the instructions provided with the c-tree package. It is found under TPEP.PRJ\CTREE\ on the source code diskettes.

7. wndowmlc.lib

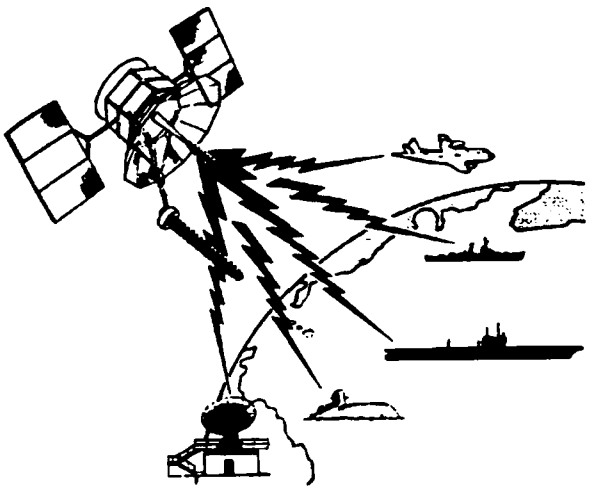
This is created by following the instructions provided with the MetaWindow Plus package. It is found under TPEP.PRJ\METAWNDW\ on the source code diskettes.

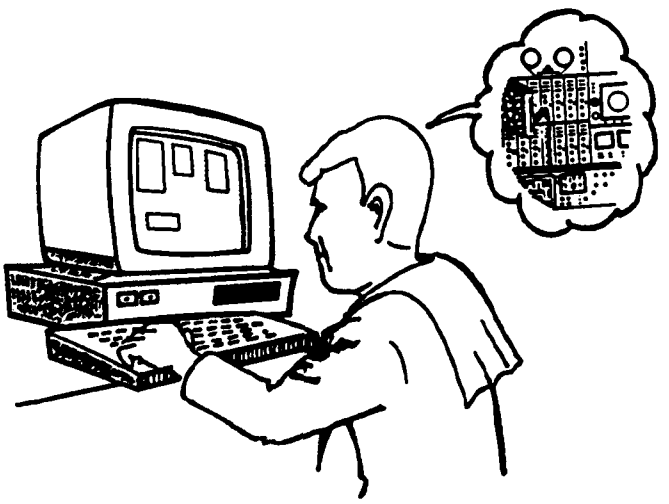
APPENDIX D

TAE ADMINISTRATION GUIDE

TAE

ADMINISTRATION GUIDE





FOREWORD

The Troubleshooting Assessment and Enhancement (TAE) program originally was an experimental troubleshooting (TS) effort investigating the feasibility of a training and evaluation program which involved the design, development, test, and evaluation of an innovative way to provide for fleet, reserve readiness center and school TS training and enhancement. The success of this test depended heavily on the involvement and support by appropriate Naval personnel, particularly the person responsible for the technical training, evaluation and development of new technical personnel (e.g., Instructor, Work Center Supervisor. Leading Petty Officer, Training Petty Officer, etc.) or the person designated as the TAE Administrator. It was acknowledged that some additional time and effort would be required on the part of the leading personnel for proper administration of the program. However, a primary design goal of TAE was to minimize the amount of administrative time and effort required through various supporting devices. The TAE guide is one such device.

The initial phase of the TAE experimental program is currently underway and will be completed with final reports. Now, based on recommendations from subject matter experts and the need by the Chief of Naval Education and Training (CNET) to provide training via new and non-traditional ways (e.g., computer assisted, "pierside training vans," etc.) the TAE program is being provided to training and operational personnel. This implementation is to provide the opportunity for utilization by all schools and ships and stations having the NAVMACS/SATCOM system requiring NEC ET-1453 qualified personnel.

In a program such as TAE, a number of requirements, conditions, and circumstances must be anticipated and provided for administratively. The Administration Guide presents information that the TAE Administrators will need to run the program effectively. The TAE Administrator will provide day-to-day administration of the program at his/her site. This is an extremely important role. The Administrator will run the delivery system, enter personnel into the program, provide guidance and counseling (as appropriate), provide program results, and assess progress of the individuals enrolled in the program.

There may be a need for additional information which might prove helpful to those personnel administering at operational sites. Therefore, we are very much interested in obtaining feedback on any aspect of the TAE program particularly as related to topics which should be added/revised and/or other recommendations for improving the general usefulness of the Guide. Please provide feedback to the TAE Project Office, Code 142 NPRDC, San Diego, CA 92152-6800.

TABLE OF CONTENTS

TRAINING ASSESSMENT AND ENHANCEMENT ADMINISTRATION

THE TAE PROGRAM

This section provides an introduction to the Troubleshooting Assessment and Enhancement (TAE) program. It describes the intent of the program, what it is trying to accomplish, the delivery system, the various troubleshooting (TS) presentations, how the evaluation/assessment scheme is intended to be utilized and how the TAE TS episodes can be used for training, assessment and improvement of individual and school performance.

T A E

training
assessment
and
enhancement

Background

Currently the Navy is faced with increasing training and operational costs and declining entry level skills. Results from other naval studies indicate that the capability of the shipboard technicians and their ability to efficiently and cost effectively maintain combat systems is of great concern. While more complex systems and equipment are increasing maintenance difficulty, workload, and costs, the personnel and time available for the training of these personnel in the maintenance and improvement of their skills are, at best, fixed. Unit and shorebased training costs continue to increase, budgets have shrunk, and the return from a heavy investment in training is being questioned. There was a need for the design, development, test & evaluation, and implementation of a low cost method to optimize school training time and to provide for fleet and reserve readiness center development and maintenance of the critical TS skill.

T A E
ADDRESSES THE PROBLEMS
OF TROUBLESHOOTING TRAINING
AND ASSESSMENT

What is TAE?

Sponsored by CNO, TAE is a TS training, assessment, and enhancement system that offers the opportunity for improving TS skill development, improvement, and maintenance. TAE has total force utility. It was initiated because the Navy has no means of measuring, for diagnostic purposes, the troubleshooting proficiency of maintenance technicians and their ability to accomplish assigned TS tasks. Other than subjective supervisory or instructor opinion, there is no way to objectively and consistently assess and diagnose technician/student performance to improve the effects of training in the Navy "C" schools as related to remediation of personnel who fall below performance criteria. Once the "C" school graduate has been integrated into the ships force, fleet commanders have no method to maintain the technicians performance capabilities to obviate skill degradation over time. In addition, the schools receive no quantifiable feedback identifying specific areas where troubleshooting training requires greater emphasis or improvement.

T A E
IMPROVES TROUBLESHOOTING
SKILL DEVELOPMENT

Due to the limited availability of systems hardware at the "C" schools actual hands-on training time is severely restricted. This minimizes the amount of time students use their system knowledge explicitly and, therefore, decreases the effectiveness of instructional programs. Once on-board, the ship safety hazards associated with corrective maintenance of weapon systems hardware preclude the use of drill and practice exercises. This limits the technicians' ability to maintain their troubleshooting skills and restricts improvement of their abilities.

TAE
INCREASES TIME STUDENTS
USE SYSTEM KNOWLEDGE

Navy Reserve personnel must receive training to achieve the same level of proficiency as the active forces. In fact, given the lessened opportunity for actual hands-on practice, the amount of reserve training should exceed that of the active forces. To achieve these objectives, innovative uses of limited training resources, particularly time, must be identified. In addition, there is no way to test and objectively assess the transfer of training or performance capabilities of the reserve personnel. At this time there is no quantifiable way to ensure comparable reserve and active duty (i.e., total force) personnel in terms of their ability to contribute to operational readiness.

Objective of TAE

This effort was initiated to develop a microcomputer-based training and assessment system which provides diagnostic capability for the remediation of personnel failing to meet criteria established for technical troubleshooting proficiency. This capability can be used in support of TS skills development, assessment, and maintenance of the total forces of the operational and training communities of the Navy.

TAE
PROVIDES DIAGNOSTIC
CAPABILITY FOR STUDENT
REMEDIATION

Operational Objective of TAE

The TAE program supports the operational Navy (total forces) and the training communities and can, within the hardware systems selected, provide for:

- Assessment of troubleshooting capabilities within the Navy training environment, better use of Technical Training Equipment (TTE) in "C" schools.
- Drill and practice for personnel under training awaiting hardware availability or active duty assignments (improved utilization of student time, and improved use of TTE).
- Curricula and training methods improvement based on feedback of school troubleshooting assessment results.
- Training through drill and practice exercises.
- Assessment of training effectiveness and retention in the skill area of troubleshooting proficiency.

TAE
PROVIDES DRILL AND
PRACTICE TROUBLESHOOTING

- A diagnostic capability for remediation to improve TS skills of operational personnel in the area of systems hardware troubleshooting.
- Readiness improvement based on diagnostic feedback with, and as a result of, diagnostic information being provided to appropriate offices.
- Improvement of curricula and instructional methods as a result of diagnostic feedback to the training community.
- A TAE troubleshooting episode development process.

TAE Technological Objective

The TAE includes advances in the technology of assessment of higher order conceptual skills (i.e., problem solving/troubleshooting,) which can in turn provide for modeling of maintenance performance. Definition of the factors that can be used for diagnostic purposes to indicate and remediate personnel for successful maintenance performance which provides for a point of departure in the development of a systematic research, development, test and evaluation effort. The objective of this effort is to ensure that training and aiding will be responsive to operational requirements of the active and reserve naval forces (current and projected) and will improve the ability of the maintenance personnel in the area of maintenance readiness. The result of this effort will provide input to or address issues such as:

- Technical information needs in maintenance performance.
- Definition of expert troubleshooting strategies.
- Specifications for job-aiding software.
- On-the-job (fleet) training delivery techniques.
- Use of troubleshooting analysis data in current Instructional System Development.
- Relationships between current pedagogical approaches and performance results.
- Development and acquisition of troubleshooting aids/trainers and devices, General Purpose Electronic Test Equipment (GPETE) and special purpose test equipment.
- Development of generic troubleshooting/problem solving model(s).

TAE Administration

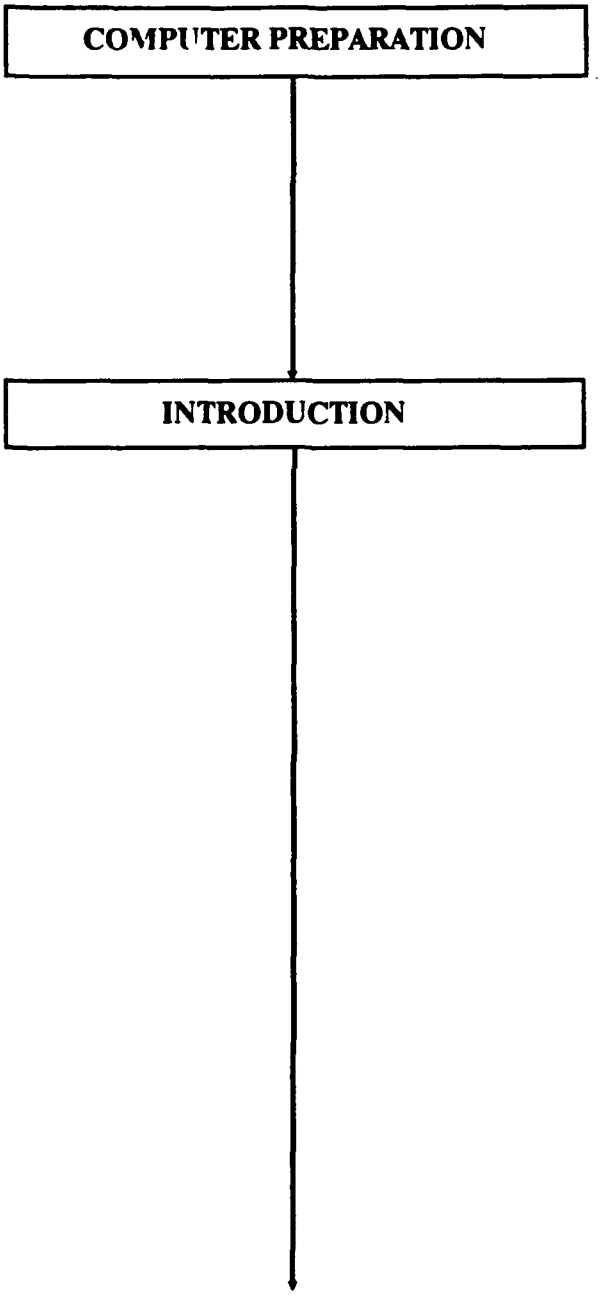
The following pages provide information on how to administer the TAE program. The procedures have been developed in a Job Performance Aid (JPA) format so that individual pages can be used, extracted, or copied in whatever way the administrator feels is most effective.

TAE
WILL IMPROVE THE ABILITY OF
MAINTENANCE PERSONNEL IN THE
ACTIVE AND RESERVE NAVAL FORCES

TAE ADMINISTRATION

OVERVIEW

The Overview is presented for the Administrator that is familiar with the Administration Process. It can be used as a quick reference.



Computer Preparation

- Turn on computer/monitor.
- Check Date and Time.
- New student.
  - Yes - Learn.
  - No - Testing.

Introduction

- Introduce self.
- Introduce organization.
- Introduction and TAE Program materials.
  - Student read or,
  - Describe.
- What Makes a Good Troubleshooter?
  - Past
  - More to it.
- TAE provides "Drill and Practice."
- NAVMACS/SATCOM first System.
- NAVMACS and NSV Equipment.
- Similar to Paper and Pencil TS.
- Installed on computer.
- Computer Tracks Steps.
- Examples of Processes Available.
  - Front Panels
  - Maintenance Panels
  - Diagnostics
  - Signals/Voltages at Test Points

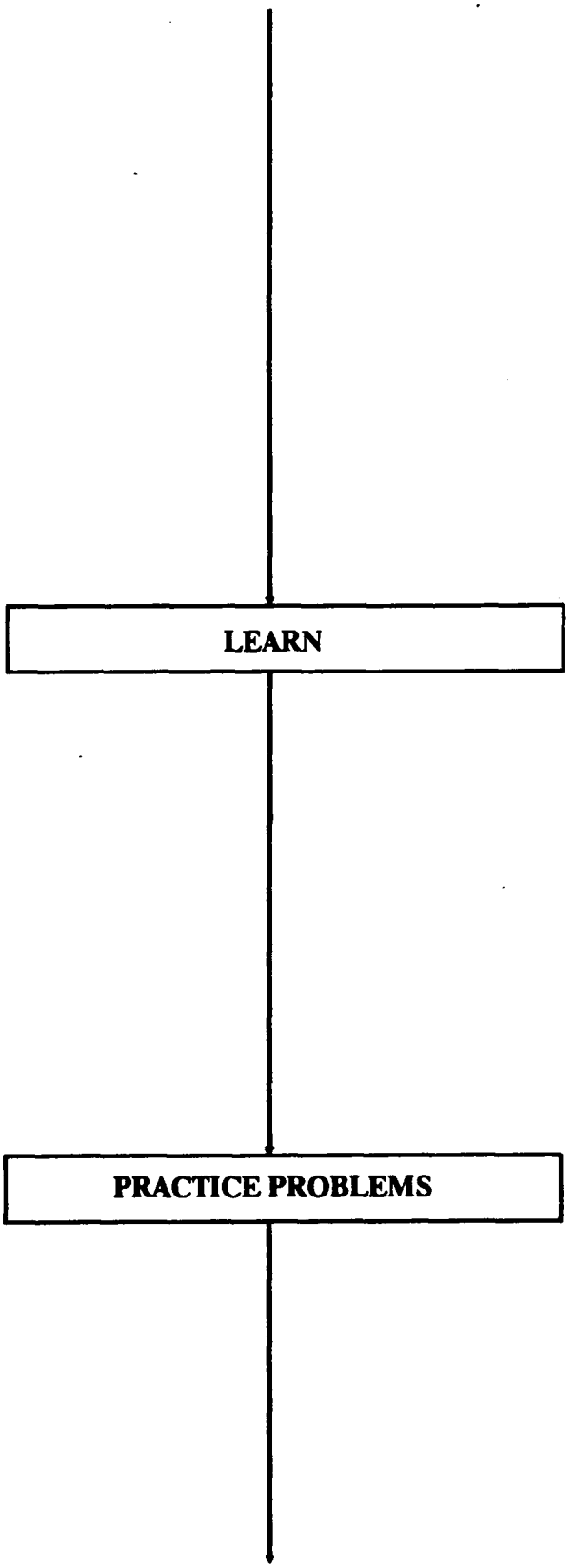
- TAE like Fleet troubleshooting.
  - NO Notes Allowed.
  - Use ONLY Official Technical Publications.
  - Troubleshoot to Lowest Replaceable Unit (LRU).
- Agenda/Sequence
  - Gather demographic information.
  - Privacy Act Statement.
  - Learn Program.
  - Two practice problems.
  - Fourteen test problems.
  - Debrief.
- Students sign Privacy Act Statement.
- Set up testing schedule.

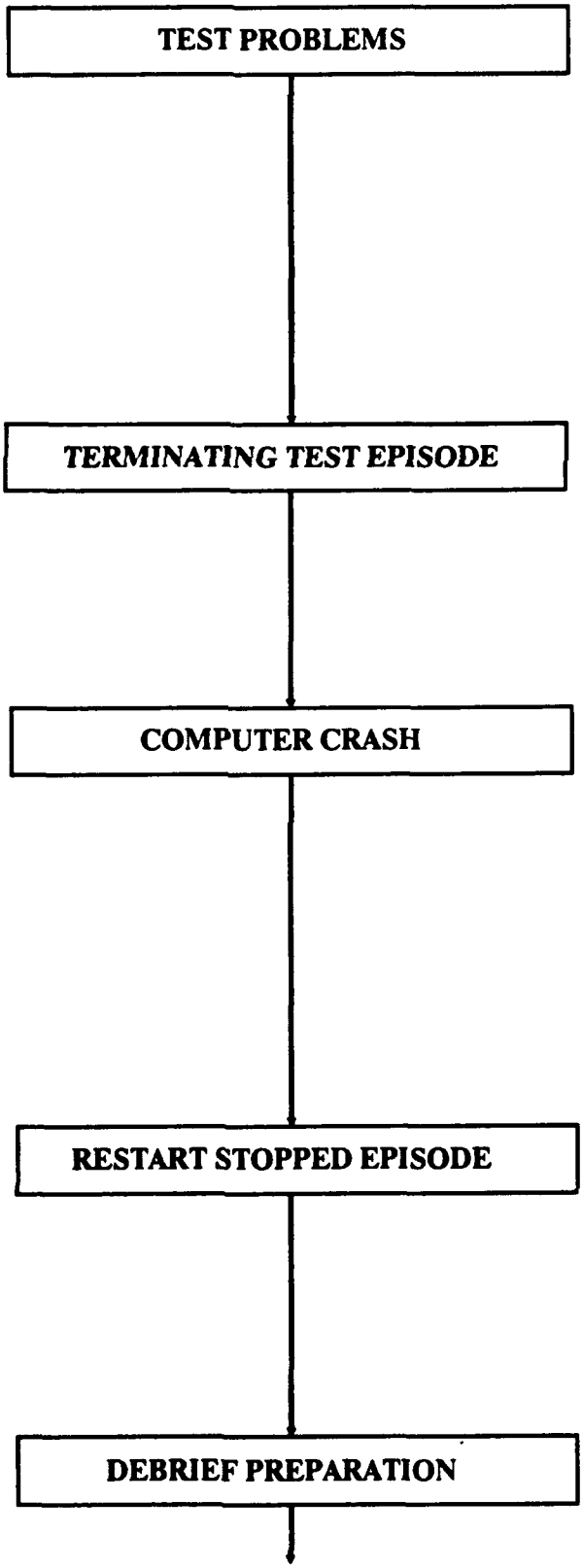
Learn

- Select "TAE Learn" program.
- Inform student.
  - Free-Running.
  - Sit and watch.
- Any questions.
- TAE Help paper.
  - Reference designation examples.
  - Step procedure examples.
  - Error messages.
- Proceed to "Practice Problems."

Practice Problems

- Select "Testing" program.
- Select first practice problem.
- Enter on "Tracking Form."
- Listen for "Beeps."
- Observe for proper entries.
- Select second practice problem.
- Enter on "Tracking Form."
- Proceed to "Test Problems."





Test Problems

- Select first problem as per "Tracking Form."
- Enter on "Tracking Form."
- Repeat for remaining test problems.
- Ensure all problems complete.
- "Non-Test" Problems.
- Proceed to "Debrief Preparation."

Terminating Test Episode

- Proceed to "System Level" menu.
  - Select "blank area."
  - Enter password.
- Enter reason.

Computer Crash

- Contact TAE Project Office.
- Record conditions.
- Re-boot or restart computer.
- Use temporary SSN.
- Redo TS Episode.
- Exit.

Restart Stopped Episode

- Record reasons for stoppage.
- Use temporary SSN.
- Redo TS Episode.
- Exit.

Debrief Preparation

- Select "Editview" program.
- Select student.
- Record student scores on "Scoring Sheet."

Debrief

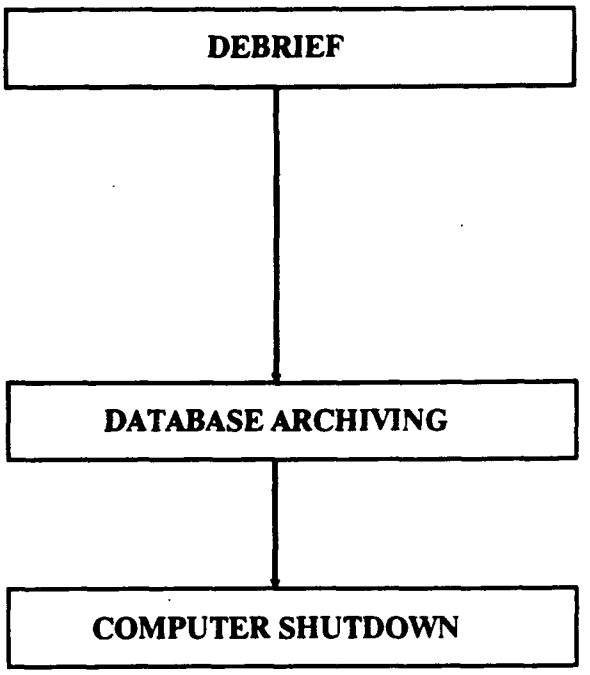
- Student completes "TAE Critique Form."
- Discuss Ranking Factors.
- Provide Scores.
- Discuss Problems.
- Thanks you.

Database Archiving

- Select "Editview" program.
- "Archive" student history.

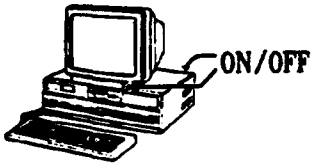
Computer Shutdown

- Select "Quit."
- Turn off computer/monitor.

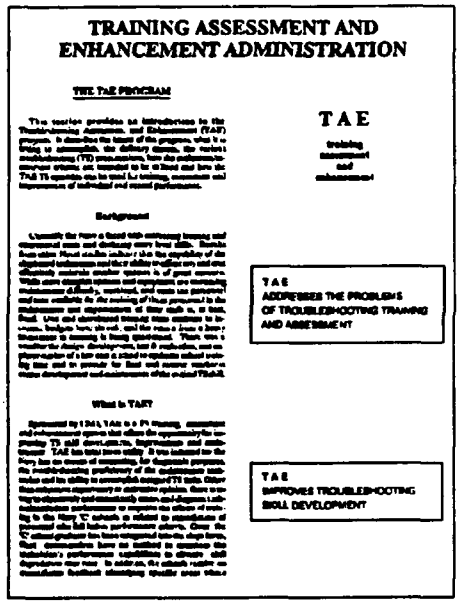


STEP PROCEDURES

COMPUTER PREPARATION



1. Turn on computer/monitor.
2. When IBM LOGO comes up:
   a. Check to ensure correct/change Date. Press <ENTER>.
   b. Check to ensure correct/change Time. Press <ENTER>.
   c. Determine if a new student -- if yes, go to "Introduction" Step 1; if no, go to "Test Problems" Step 1.



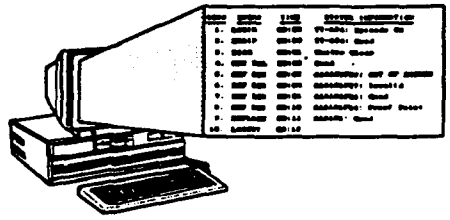
INTRODUCTION

1. Introduce yourself and any assistants.
2. Introduce your organization and what is the intent of this testing.
3. Introduce the TAE Program and the associated materials by:
   a. Giving a lecture or
   b. Having student read the TAE "introduction" and "background" information presented in this TAE ADMINISTRATION HANDBOOK.
4. Discuss "what makes a good troubleshooter"; also explain:
   a. In the past determining what makes a "good troubleshooter was based on -- finding the problem, available time, and the number of tests.
   b. Explain that this approach takes into account approach used, out of bounds, time between tests, repeat steps, type of tests, etc.
5. Also explain that the program can be used for "Drill and Practice" to maintain skills in troubleshooting once testing is completed.
6. Explain that the first system selected for design, development, test, and evaluation was the NAVMACS/SATCOM system.
7. Ensure that they (the students) are aware that the NAVMACS and NSV Equipment are covered.
8. Describe "generally" the process; explain that the approach is similar to "paper and pencil" troubleshooting exercises.

1ST SYSTEM FOR TEST

9. Tell them that the difference is that the troubleshooting activities are recorded and "put on computer" so that the results can be assessed to improve TS training and individual skills i.e., TS abilities can be "enhanced."

10. The computer "Tracks the Steps" used so that the approaches to troubleshooting can be analyzed to develop a "model" of troubleshooting, which will be used to improve the course of instruction on the system.



11. Identify "Examples of Processes Available" such as:

    a. Front Panels.

    b. Maintenance Panels.

    c. Diagnostics.

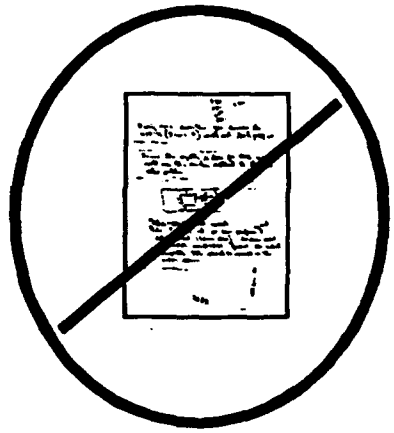
    d. Signals at Test Points.

12. Ensure that they understand that the approach is set up to provide TS exercises "Like Fleet Troubleshooting"; i.e.:

    a. No school or personal notes allowed.

    b. Use ONLY "official" technical manuals/documentation (Appendix D/ A-1).

    c. Troubleshooting will be to the Lowest Replaceable Unit (LRU).



13. Describe the "Agenda/Sequence" of the TS exercise; i.e.:

    a. Gathering of Demographic Data.

    b. Filling out of Privacy Act Statement.

    c. Watch Learn Program.

    d. Do two Practice Problems.

    e. Do 14 Test Problems.

    f. Debrief.

    g. Assign more problems if time available, after test problems done.

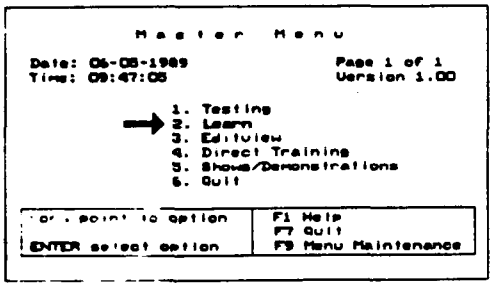
Testing Schedule

STUDENT	COMPUTER	TIME
PO3 Dudley	A	10:00
PO2 Conner	B	10:00
PO3 Baker	A	11:15
PO3 Smith	B	11.15

14. Gather as much demographic information as you can, using the "Service Information" sheet (Appendix D/ A-2).

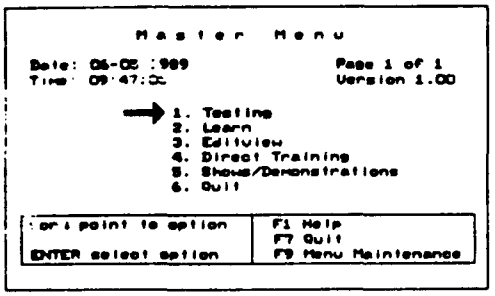
15. Have student/sailor read and sign the Privacy Act Statement (Appendix D/ A-3).

16. Set up Testing Schedule, based on student and computer availability.



LEARN

1. Have student take the Learn Program.
   a. From "Master Menu" (using up and down arrow keys) Select: "LEARN."
   b. Press <ENTER>.
2. Inform the student that all is required is to pay attention to the presentation.
   a. The program is "free running."
   b. Just "sit and watch."
   c. Have him/her contact you when program is finished.
3. Present the "Help Paper" (Appendix D/ A-4)
   a. Show and explain the "Reference Designation" examples and how important it is to enter the RefDes exactly.
   b. Show and explain the "Step Procedure" examples and that only "decision" steps are necessary; i.e., the assumption is that he/she can flip switches, turn knobs etc.
   c. Explain what the error message i.e., out-of-bounds means.
4. Upon completion answer any questions concerning the program.
5. Upon completion of Learn Program:
   a. Press "Esc" key.
   b. At "Show Terminated," press <ENTER>.
6. Go to "Practice Problems" Step 1.



PRACTICE PROBLEMS

1. Have student take the two practice problems.
   a. From "Master Menu" (using arrow keys) Select: "Testing."
   b. Press <ENTER>.
2. From "Administrators Menu":
   a. Using arrow keys, select "Test Episodes."
   b. Press <ENTER>.

3. When asked, enter Password ( ).

4. When asked, enter Student SSN and press <ENTER>.

5. From the "TAE Episode Testing Sequence and Recording" Form (Appendix D/ A-5):

   a. Determine/Select the proper Column to use. Column 1 for SSNs ending in odd numbers and Column 2 for SSNs ending in even numbers.

   b. Determine/select the practice problem Sub-system

   c. Determine/select the Episode Number of the Sub-system.

Example: **Odd number SSN - Column 1.**

1st Practice Episode - 1. 26 #4.

NOTE: Ensure that you track and note episodes as they are taken and continue to have the student accomplish episodes as listed.

6. From "Episode Database" menu:

   a. Using arrow keys, select Appropriate Sub- system [e.g., AN/USH- 26(V)].

   b. Press <ENTER>.

7. From the selected Sub-system menu; e.g., "AN/USH-26(V)" Episodes" menu:

   a. Using arrow keys, select Appropriate Episode (e.g., 4. Parallel Interface).

   b. Press <ENTER>.

8. When told "Episode Loaded," press <ENTER>.

9. Have Student commence/accomplish TS Episode.

10. Record TS Episode taken on "Tracking Form."

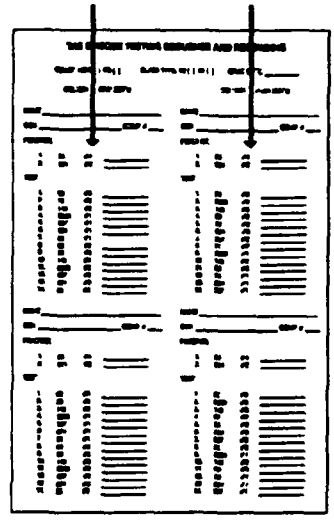
NOTE: Listen for an inordinate number of "beeps" from the computer as the student goes through the episode. This may indicate that the student is entering the reference designators, step procedures, etc. improperly. Ensure that he/she has and is attending to the "TAE TS Episode Reference Designator Entries And Error Messages" sheet.

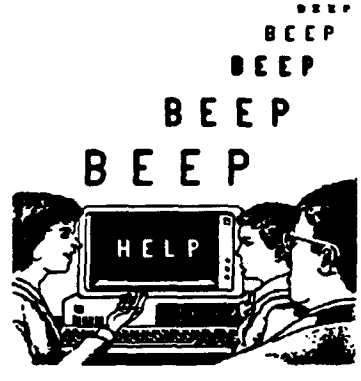
NOTE: If it is necessary to terminate the episode prior to completion, go to procedure "Terminating an Episode Before Completion."

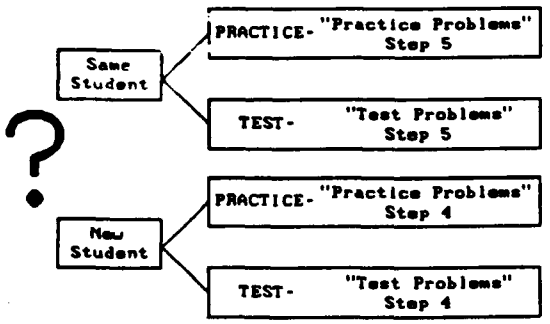
ODD SSN **EVEN SSN**

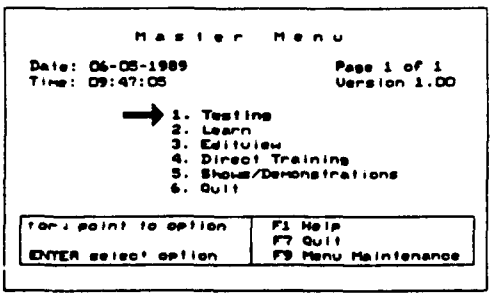
USE **USE**

COLUMN 1 **COLUMN 2**









11. Upon completion of the Episode by the student and upon the message "Congratulations!! Your Replacement Solves the Problem"; press <ENTER>.

12. When asked, enter Password ( ).

13. From "Student" Menu (using arrow keys), select:

    a. "Test on Same Student" and press <ENTER> -- if continuing Practice Problems on the same student, return to "Practice Problems" Step 5.

    b. "Test on Same Student" and press <ENTER> -- if going to Test Problems on the same student, go to "Test Problems" Step 5.

    c. "Test on New Student" and press <ENTER> -- if going run Practice Problems on the new student, return to "Practice Problems" Step 4.

    d. "Test on New Student" and press <ENTER> -- if going to Test Problems on the new student, go to "Test Problems" Step 4.

    c. "Exit" and press <ENTER> -- if terminating testing session.

TEST PROBLEMS

1. *Have student take the 14 test problems.*

    a. From "Master Menu" (using arrow keys) Select: "Testing."

    b. Press <ENTER>.

2. From "Administrators Menu":

    a. Using arrow keys, select "Test Episodes."

    b. Press <ENTER>.

3. When asked, enter Password ( ).

4. When asked, enter Student SSN and press <ENTER>.

5. Continue selecting the TS problems from the "TAE Episode Testing Sequence and Recording" Form used in the practice problems.

    a. Determine/select the practice problem Sub-system

    b. Determine/select the Episode Number of the Sub-system.

Example: **Even number SSN - Column 2.**
1st Test Episode - 1. 20 #3.

NOTE: Ensure that you track and note episodes as they are taken and continue to have the student accomplish episodes as listed. Once all the "Test And Evaluation" episodes have been completed, you may present the remainder in any sequence/number you wish.

6. From "Episode Database" menu:

   a. Using arrow keys, select Appropriate Sub- system [e.g., AN/UYK-20(V)].

   b. Press < ENTER >.

7. From the selected Sub-system menu; e.g., "AN/UYK-20(V)" Episodes" menu:

   a. Using arrow keys, select Appropriate Episode (e.g., 3. Channel 14 Interface).

   b. Press < ENTER >.

8. When told "Episode Loaded," press < ENTER >.

9. Have Student Commence/accomplish TS Episode.

10. Record TS Episode taken on "Tracking Form."

NOTE: Listen for an inordinate number of "beeps" from the computer as the student goes through the episode. This may indicate that the student is entering the reference designators, step procedures, etc. improperly. Ensure that he/she has and is attending to the "TAE TS Episode Reference Designator Entries And Error Messages" sheet.

NOTE: If it is necessary to terminate the episode prior to completion, go to procedure "Terminating an Episode Before Completion."

11. Upon completion of the Episode by the student and upon the message "Congratulations!! Your Replacement Solves the Problem," press < ENTER >.

12. When asked, enter Password ( ).

13. From "Student" Menu (using arrow keys), select:

   a. "Test on Same Student" and press < ENTER > -- if continuing Test Problems on the same student, return to "Test Problems" Step 5.

   b. "Test on New Student" and press < ENTER > -- if going run Practice Problems on the new student, go to "Practice Problems" Step 4.

Select Sub-System
AN/USH-26(V)
AN/USQ-69(V)
AN/UYK-20(V)
CV-3333/U
ON-143(V)/USQ
RD-397(V)/U
TT-624(V)UG

Record Each Episode

c. "Test on New Student" and press < ENTER > -- if going to Test Problems on the new student, go to "Test Problems" Step 4.

d. "Exit" and press < ENTER > -- if terminating testing session.

TERMINATING TEST EPISODE

The following steps are to be used, if a need arises, to END a TS Episode before the student is done.

1. Return to "System Level" Menu. Use following procedure until the "System Level" Menu is reached.

   a. Using arrow keys, select Return.

   b. Press < ENTER >.

2. From "System Level" Menu.

   a. Using arrow keys, select the "blank area under Replace LRU."

   b. Press < ENTER >.

3. When asked, enter Password ( ).

4. ENTER the reason for the termination of the episode and who okayed it.

   a. TYPE your initials, a blank space, and short reason for termination.

   b. Press < ENTER >.

Example: DED Go to Dental.

5. From "Student" Menu (using arrow keys), select:

   a. "Test on Same Student" and press < ENTER > -- if continuing Test Problems on the same student, return to "Test Problems" Step 5.

   b. "Test on New Student" and press < ENTER > -- if going run Practice Problems on the new student, go to "Practice Problems" Step 4.

   c. "Test on New Student" and press < ENTER > -- if going to Test Problems on the new student, go to "Test Problems" Step 4.

   d. "Exit" and press < ENTER > -- if terminating testing session.

Terminate

An Episode

Before

Completion

For

Sickness,

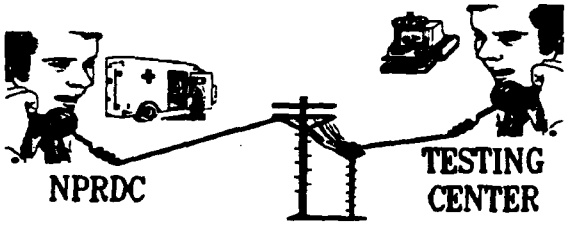
P T,

Etc.

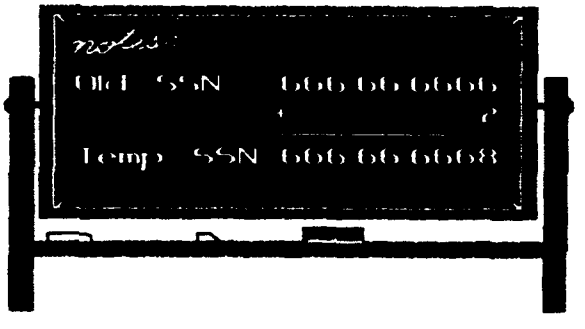
COMPUTER CRASH

The following steps should be performed if computer locks-up or some other event occurs so that the computer quits working.

NOTE: Contact the TAE Project Office (A/V 553-7665 or A/V 553- 7664) and provide information as to what occured to cause, and the results of the, "crash."



1. Record on the "TAE Episode Testing Sequence and Recording" Sheet:
   a. "Exactly" the conditions that exist.
   b. What the computer was presenting when it stopped.
2. Attempt "HOT" re-boot.
   a. Hold the "Ctrl" and "Alt" keys down and
   b. Hit the "Del" key.
3. If Step 2 does not work:
   a. Attempt a "total" restart by turning off the computer.
   b. Wait two minutes before going to Step 4.
4. Turn on computer.
5. Have student complete Test Episode, when the computer starting acting up.
   a. From "Master Menu" (using arrow keys), select: "Testing."
   b. Press <ENTER>.
6. From "Administrators Menu":
   a. Using arrow keys, select "Test Episodes."
   b. Press <ENTER>.
7. When asked, enter Password ( ).
8. When asked to enter Student SSN:
   a. Add two (2) to the last digit of the students SSN.



Examples: If SSN is 222-22-2222, it would become 222-22-2224.

If SSN is 888-88-8888, it would become 888-88-8880.

b. Type in this new temporary SSN.

c. Press <ENTER>.

9. From "Episode Database" menu:

   a. Using arrow keys, select Appropriate Sub- system to redo.

   b. Press <ENTER>.

10. From the selected Sub-system menu " Episodes" menu:

    a. Using arrow keys, select Appropriate Episode to redo.

    b. Press <ENTER>.

11. When told "Episode Loaded," press <ENTER>.

12. Have Student Commence/Accomplish TS Episode, when the computer starts acting up.

13. Upon completion of THIS Episode by the student and upon the message "Congratulations!! Your Replacement Solves the Problem," Press <ENTER>.

14. When asked, enter Password ( ).

15. From "Student" Menu (using arrow keys), select:

    a. "Test on New Student" and press <ENTER> -- if going to Test Problems on the SAME student, go to "Test Problems" Step 4 and enter correct SSN.

    b. "Test on New Student" and press <ENTER> -- if going to Test Problems on the new student, go to "Test Problems" Step 4.

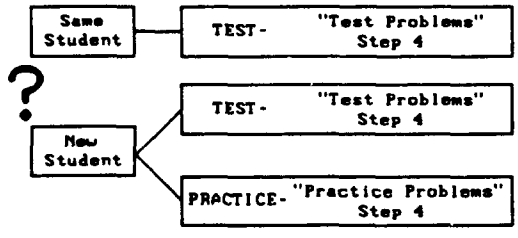
    c. "Test on New Student" and press <ENTER> -- if going to run Practice Problems on the new student, go to "Practice Problems" Step 4.

    d. "Exit" and press <ENTER> -- if terminating testing session.

SELECT SAME

SUB-SYSTEM

AND EPISODE



RESTARTING STOPPED EPISODE

The following steps should be performed if a student needs to be restarted on a TS Episode that needed to be stopped for whatever reason.

1. Record on the "TAE Episode Testing Sequence and Recording" Sheet the reason for the stoppage.

2. Have student complete the Test Episode that was stopped.

   a. From "Master Menu" (using arrow keys) Select: "Testing."

b. Press <ENTER>.

3. From "Administrators Menu":

   a. Using arrow keys, select "Test Episodes."

   b. Press <ENTER>.

4. When asked, enter Password ( ).

5. When asked to enter Student SSN:

   a. Add two (2) to the last digit of the students SSN.

Examples: **If SSN is 555-55-5555, it would become 555-55-5557.**

If SSN is 999-99-9999, it would become 999-99-9991.

REMEMBER

At End Of A Restarted Episode, Enter Students Correct SSN.

   b. Type in this new temporary SSN.

   c. Press <ENTER>.

6. From "Episode Database" menu:

   a. Using arrow keys, select Appropriate Sub- system to redo.

   b. Press <ENTER>.

7. From the selected Sub-system menu " Episodes" menu:

   a. Using arrow keys, select Appropriate Episode to redo.

   b. Press <ENTER>.

8. When told "Episode Loaded"; Press <ENTER>.

9. Have Student commence/accomplish TS Episode, when the computer starts acting up.

10. Upon completion of THIS Episode by the student and upon the message "Congratulations!! Your Replacement Solves the Problem," press <ENTER>.

11. When asked, enter Password ( ).

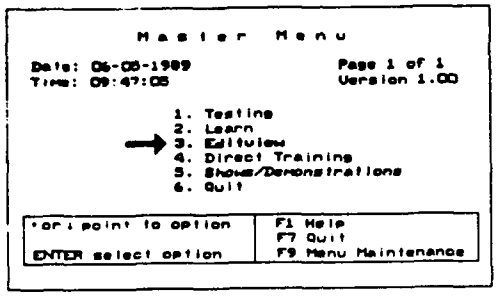
12. From "Student" Menu (using arrow keys) select:

   a. "Test on New Student" and press <ENTER> -- if going to Test Problems on the SAME student, go to "Test Problems" Step 4 and enter correct SSN.

   b. "Test on New Student" and press <ENTER> -- if going to Test Problems on the new student, go to "Test Problems" Step 4.

c. "Test on New Student" and press <ENTER> -- if going to run Practice Problems on the new student, go to "Practice Problems" Step 4.

d. "Exit" and press <ENTER> -- if terminating testing session.



DEBRIEF PREPARATION

The following steps should be performed to collect and record students scores on all 14 TS episodes.

1. From "Master Menu":

   a. Using arrow keys, select "EDITVIEW."

   b. Press <ENTER>.

2. From "Main Menu":

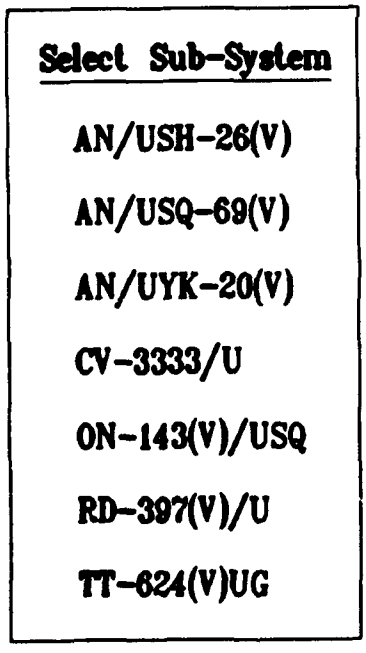
   a. Using arrow keys, select "EXAMINE Individual History."

   b. Press <ENTER>.

3. From the List of Students in Database (as shown on "Students Currently in Database"):

   a. Enter desired students SSN.

   b. Press <ENTER>.



4. From "Episode Database" menu:

   a. Using arrow keys, select Appropriate Sub- system to be scored [e.g., AN/USH-26(V)].

   b. Press <ENTER>.

5. From the selected Sub-system menu; e.g., "AN/USH-26(V)" Episodes" menu:

   a. Using arrow keys, select Appropriate Completed (DONE) Episode to be scored (e.g., Formatter A).

   b. Press <ENTER>.

6. From "Individual History" Menu:

   a. Using the arrow keys, select "DISPLAY ON SCREEN."

   b. Press <ENTER>.

7. From "DISPLAY" Menu:

   a. Using the arrow keys, select "Scored Summary."

   b. Press <ENTER>.

8. From "Scoring Summary" screen (Appendix D/ A-6):

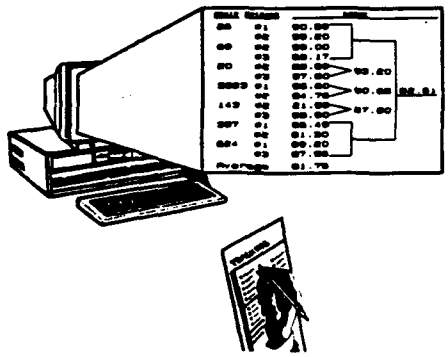
   a. Record on Student Score sheet the students score for the TS Episode being scored.

   b. Press <ENTER>.

9. From "Display" Menu:

   a. Using the arrow keys, select "Return."

   b. Press <ENTER>.



10. From "Individual History" Menu:

    a. Using the arrow keys, select "Return."

    b. Press <ENTER>.

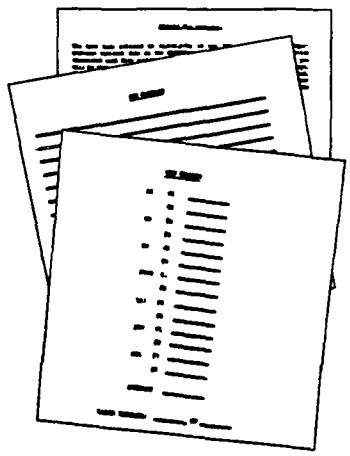
11. From "Student Menu" select:

    a. "Examine History: Same Student" and press <ENTER> -- to continue scoring on student, go to "Debrief Preparation" Step 4.

    b. "Examine History: New Student" and press <ENTER> -- to score on a new student, go to "Debrief Preparation" Step 3.

    c. "Exit" and press <ENTER> -- if terminating scoring session.

DEBRIEF



1. "TAE Critique" Form (Appendix D/ A-7).

   a. Have student complete form providing feedback as to his/her perceptions of the program good/bad/indifferent and any recommendations for improvement/modification.

   b. Collect "TAE Critique" Form.

2. "Ranking Factors" Sheet.

   a. Hand out a copy of the "Ranking Factors" (Appendix D/ A-8).

   b. Discuss how they are used to determine scores.

3. "Scored Episode" Sheet.

   a. Hand out a copy of the "Scored Episode" (Appendix D/ A-9).

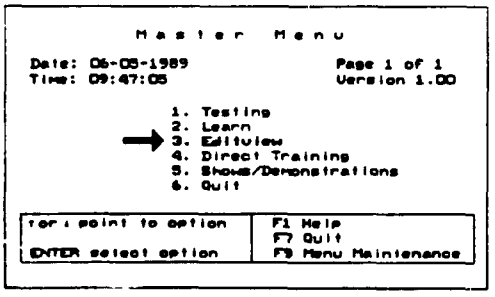
   b. Discuss how scoring is accomplished.

Discuss

Problems

Answer

Questions



4. "Sequence Of Events" Sheet.

   a. Hand out a copy of the "Sequence Of Events" (Appendix D/ A-10).

   b. Discuss how the computer tracks every move.

5. "Student Score" Sheet.

   a. Hand out "Student Score" Sheets to each student.

   b. Discuss, if necessary.

6. Collect "Ranking Factors," "Scored Episode," and "Student Score" Sheets.

7. Discuss problems and answer any questions.

   a. Discuss the problems showing how one could have used the publications more effectively.

   b. Answer any questions that the students might have.

8. Provide "thank you's" for their assistance.

DATABASE ARCHIVING

1. From "Master Menu":

   a. *Using arrow keys,* select "EDITVIEW."

   b. Press <ENTER>.

2. From "Main Menu":

   a. Using arrow keys, select the blank space below "EXIT."

   b. Press <ENTER>.

3. When asked, enter Password ( ).

4. From "Administrative" Menu:

   a. Using arrow keys, Select "select Database Archiving."

   b. Press <ENTER>.

5. From "Archiving Functions" Menu:

   a. Using arrow keys, Select "Archive a student history database."

   b. Press <ENTER>.

6. From "Set Archive Name" Menu:

   a. Using arrow keys, select "Set Location."

b. Press <ENTER>.

7. From "Set Location" Menu:

   a. Using arrow keys, select "San Diego" and press <ENTER> -- if testing on the west coast.

   b. Using arrow keys, select "Norfolk" and press <ENTER> -- if testing on the east coast.

8. From "Set Archive Name" Menu:

   a. Using arrow keys, select "Set Date."

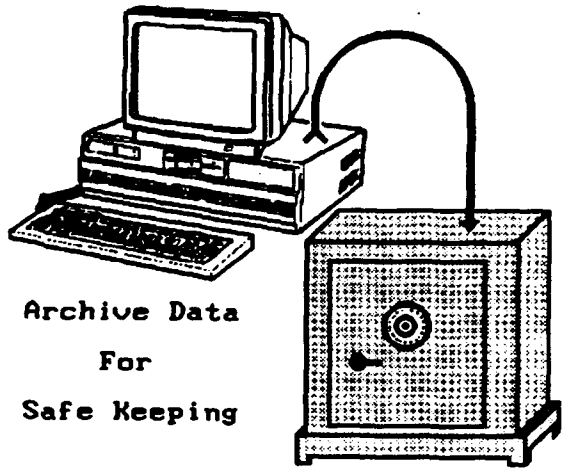
   b. Press <ENTER>.

9. When asked to enter date:

   a. Type in "MMDDYY" and Press <ENTER> -- if testing in school, enter course completion date.

   b. Type in "MMDDYY" and Press <ENTER> -- if individual or ship testing, enter todays date.

Example - 032889



10. From "Set Archive Name" Menu:

    a. Using arrow keys, select "Proceed with Archiving."

    b. Press <ENTER>.

11. When told "Archiving Complete" -- Press any key.

12. From "Archiving Functions" Menu.

    a. Using arrow keys, select "Return to administrative menu."

    b. Press <ENTER>.

COMPUTER SHUTDOWN

1. Return to "Master Menu."

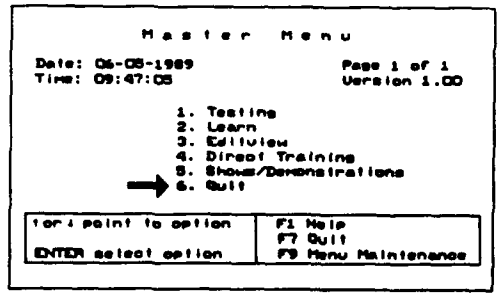
2. From "Master Menu."

   a. Using arrow keys, select "Quit."

   b. Press <ENTER>.

   c. Wait approximately 10 seconds before continuing.

3. Turn off computer/monitor.



APPENDIX D/ A

TAE

ADMINISTRATION FORMS

REQUIRED NAVMACS PUBLICATIONS

USH-26	NAVSEA SE640-AF-MMO-010/USH-26(V)
	NAVSEA SE640-AF-MMO-020/USH-26(V)
	NAVELEX 0967-LP-598-5030
USQ-69	NAVSEA 0967-LP-598-4010
	NAVSEA 0967-LP-598-4030
UYK-20	SE610-AV-MMO-010
	SE610-AV-MMO-050
CV-3333	Audio-Digital Converter CV-3333
ON-143	NAVELEX 0967-LP-614-7010
RD-397	SPAWAR 0967-LP-614-6010
TT-624	EE161-NA-OMI-010/E110-TT624
	NAVELEX 0967-LP-544-0050

TAE SERVICE INFORMATION

NAME (Last, First, MI) _____ DATE OF BIRTH/AGE _____ SSN _____

RATE _____ DATE OF RATE _____ TIME IN SERVICE _____ NEC #1 _____ NEC #2 _____ NEC #3 _____

APQT	GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE

TAE SCHOOL INFORMATION

SAN DIEGO [ ] V3 [ ] CLASS # _____ GRAD DATE _____
NORFOLK [ ] V2 [ ]

EQUIPMENT	QUIZ AVG	PT'S		%	TEST SCORE	AREA TOTAL
UYK20	_____	_____ OF	_____	_____	_____	_____
USH26 USQ69 RD397 TT624	_____	_____ OF	_____	_____	_____	_____
CV3333	_____	_____ OF	_____	_____	_____	_____
ON143	_____	_____ OF	_____	_____	_____	_____
KG36	_____	_____ OF	_____	_____	_____	_____
SYSTEMS	_____	_____ OF	_____	_____	_____	_____
FINAL TEST					_____	_____

CLASS STANDING _____ OF _____ FINAL SCORE _____

TAE PRIVACY ACT STATEMENT

You have been selected to participate in the TAE project. This project provides research data on the different levels of troubleshooting expertise associated with Navy systems and equipment. The information provided by you will be used by the Navy Personnel Research and Development Center, San Diego, for research purposes only. It will not become a part of your official record, nor will it be used to make decisions about you which will affect your career in any way. Your name, SSN are necessary only to aid in processing the research data.

Print Name: _____ SSN: _____

Signature: _____ Date: _____

TAE HELP PAPER

TAE TS EPISODE REFERENCE DESIGNATOR ENTRIES AND ERROR MESSAGES

REFERENCE DESIGNATION EXAMPLES

	ON-143	RD-397	TT-624
EQUIPMENT TEST POINTS	1A9TP18	1A1TP14	A1A1A13TP-A
CARD PINS	1A16P65	1A6P58	A1A1A7P33
BLACK TEST POINTS	23		
RED TEST POINTS	2		

STEP PROCEDURE EXAMPLES

	USH-26	USQ-69
STEP CATEGORY	6.3.7	10-12
STEP # OR LETTER	2	B

NOTE: **Ensure that the number 1 is entered when called for -- NOT a lower case "L". Ensure that the number 0 is entered when called for -- NOT the letter "O".**

ERROR MESSAGES

OUT OF BOUNDS - Problem is NOT in this equipment and the reading at this point is good OR,

Test Points (Card Pins) NOT in the path of the problem and the reading at this point is good OR,

Step Procedure is not in the path of the steps required to troubleshoot this problem.

ILLOGICAL APPROACH - You could troubleshoot this equipment to find the trouble, but with the symptoms given this is NOT the best approach.

INVALID CHECK - You are at a good test point, but the check you are attempting is NOT a proper check for this test point.

TAE EPISODE TESTING SEQUENCE AND RECORDING

COAST: NOR [ ] SD [ ] CLASS TYPE: V2 [ ] V3 [ ] GRAD DATE: _____

COLUMN 1 (Odd SSN's)

NAME _____

SSN _____ COMP # ____

PRACTICE

1.	26	#4	_____
2.	624	#2	_____

TEST

1.	69	#3	_____
2.	20	#2	_____
3.	143	#2	_____
4.	3333	#1	_____
5.	397	#1	_____
6.	624	#1	_____
7.	26	#1	_____
8.	69	#2	_____
9.	20	#3	_____
10.	143	#3	_____
11.	3333	#2	_____
12.	397	#2	_____
13.	624	#3	_____
14.	26	#2	_____

NAME _____

SSN _____ COMP # ____

PRACTICE

1.	26	#4	_____
2.	624	#2	_____

TEST

1.	69	#3	_____
2.	20	#2	_____
3.	143	#2	_____
4.	3333	#1	_____
5.	397	#1	_____
6.	624	#1	_____
7.	26	#1	_____
8.	69	#2	_____
9.	20	#3	_____
10.	143	#3	_____
11.	3333	#2	_____
12.	397	#2	_____
13.	624	#3	_____
14.	26	#2	_____

COLUMN 2 (Even SSN's)

NAME _____

SSN _____ COMP # ____

PRACTICE

1.	26	#4	_____
2.	624	#2	_____

TEST

1.	20	#3	_____
2.	3333	#2	_____
3.	143	#3	_____
4.	69	#2	_____
5.	26	#2	_____
6.	624	#3	_____
7.	397	#2	_____
8.	20	#2	_____
9.	3333	#1	_____
10.	143	#2	_____
11.	69	#3	_____
12.	26	#1	_____
13.	624	#1	_____
14.	397	#1	_____

NAME _____

SSN _____ COMP # ____

PRACTICE

1.	26	#4	_____
2.	624	#2	_____

TEST

1.	20	#3	_____
2.	3333	#2	_____
3.	143	#3	_____
4.	69	#2	_____
5.	26	#2	_____
6.	624	#3	_____
7.	397	#2	_____
8.	20	#2	_____
9.	3333	#1	_____
10.	143	#2	_____
11.	69	#3	_____
12.	26	#1	_____
13.	624	#1	_____
14.	397	#1	_____

TAE STUDENT SCORES

NAME:_____

26	#1	_____
	#2	_____
69	#2	_____
	#3	_____
20	#2	_____
	#3	_____
3333	#1	_____
	#2	_____
143	#2	_____
	#3	_____
397	#1	_____
	#2	_____
624	#1	_____
	#3	_____

AVERAGE _____

CLASS RANKING: _____ OF _____

TAE CRITIQUE

NAME: _____ DATE: _____
(Please Print)

Use other side for additional comments.

Ranking Factors

• SOLUTIONS	Identification of fault source/component.
• TIME	Time it takes to isolate and identify the fault.
• COST	Number of components incorrectly identified as the fault source.
• PROOF POINTS	Number of possible input and output points of faulty circuit to isolate the faulty component.
• OUT OF BOUNDS	Number of test points selected that were not relevant to diagnosing the fault.
• TEST POINTS	Number of test points, card pins, and terminal board pins examined to isolate the fault.
• VALID CHECKS	Number of valid checks (continuity, logic, frequency, current, voltage, and waveform) made to isolate the fault.
• INVALID CHECKS	Number of invalid checks (checks that cannot be made) performed at any test point.
• ILLOGICAL APPROACH	Troubleshooting testing begins at a point not indicated by the symptoms. The subject may still diagnose the fault, but did not efficiently utilize the symptom data.

Scored Episode

		100.00%
FOUND SOLUTION:	YES	0.00
TEST POINTS:	5	3.94
OUT OF BOUNDS:	1	1.09
CHECKS:	5	3.40
INVALID CHECKS:	1	1.46
ILLOGICAL APPROACH:	0	0.00
INCORRECT SOLUTION:	0	0.00
SOLUTION TEST POINTS:	2/3	3.25
TOTAL TIME:	32	7.82
SCORE:		79.04%

Sequence Of Events

EVENT	TIME	CHECKS
LOGIN	0000	
EQUIPMENT	0006	TT-624(V)UG
REF DES TEST	0011	A1A1A7P24 : WAVEFORM : GOOD
REF DES TEST	0013	A1A1A7P16 : LOGIC : INVALID
REF DES TEST	0014	A1A1A8P32 : OUT OF BOUNDS
REF DES TEST	0020	A1A1A7P14 : WAVEFORM : GOOD
REF DES TEST	0023	A1A1A7P09 : WAVEFORM : GOOD PROOF POINT 1
REF DES TEST	0026	A1A1A7P10 : WAVEFORM : GOOD PROOF POINT 3
REPLACE LRU	0032	A1A1A7 : GOOD
LOGOUT	0032	

DISTRIBUTION LIST

Distribution:
Director, Total Force Training and Education (OP-11)
Director, Training Technology (Code N-54)
Commanding Officer, Naval Education and Training Program Management Support Activity (Code 03)
Chief of Naval Technical Training (Code 00) (2)
Commander, Training Command, U.S. Atlantic Fleet
Commanding Officer, Service School Command, San Diego, CA
Defense Technical Information Center (2)

Copy to:
Commander, Naval Reserve Force, New Orleans, LA
Department of the Air Force, DET 5, Armstrong Laboratory Directorate, Brooks Air Force Base, TX
Commander, U.S. ARI, Behavioral and Social Sciences, Alexandria, VA (PERI-POT-I)
Superintendent, Naval Postgraduate School
Center for Naval Analyses